

Series acceleration formulas obtained from experimentally discovered hypergeometric recursions

Paul Levrie, John Campbell

▶ To cite this version:

Paul Levrie, John Campbell. Series acceleration formulas obtained from experimentally discovered hypergeometric recursions. 2022. hal-03659300v1

HAL Id: hal-03659300 https://hal.science/hal-03659300v1

Preprint submitted on 4 May 2022 (v1), last revised 20 Dec 2022 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Series acceleration formulas obtained from experimentally discovered hypergeometric recursions

Paul Levrie John M. Campbell

Abstract

In 2010, Kh. Hessami Pilehrood and T. Hessami Pilehrood introduced generating function identities used to obtain series accelerations for values of Dirichlet's β function, via the Markov–Wilf–Zeilberger method. Inspired by these past results, together with related results introduced by Chu et al., we introduce a variety of hypergeometric recurrences that we have discovered experimentally, and we prove these recurrences using the WZ method, and apply these recurrences to obtain series acceleration identities, including a family of summations generalizing a series for $\frac{1}{\pi^2}$ due to Guillera, and a family of summations generalizing an accelerated series for Catalan's constant due to Lupaş. We also provide "non-computer" proofs of our hypergeometric transforms.

1 Introduction

In the recent article [4], the hypergeometric transform

$$_{3}F_{2}\begin{bmatrix} a & b & 1 \\ c & 2 \end{bmatrix}; 1 = \frac{1-c}{(a-1)(b-1)} + \frac{c-1}{(a-1)(b-1)} {}_{2}F_{1}\begin{bmatrix} a-1 & b-1 \\ c-1 \end{bmatrix}; 1$$
 (1)

was proved using the Wilf–Zeilberger method [28] and was applied to formulate a new proof of Euler's formula $\zeta(2) = \pi^2/6$. The identity in (1) was also used in relation to some problems on binomial-harmonic sums given in [29]. In this paper, we provide a number of hypergeometric identities resembling (1) that were obtained experimentally. We prove them with the Wilf–Zeilberger method [28], and use them to formulate and apply series acceleration methods.

Our methods are related to Catalan's [7] and Fabry's work [12] on series evaluations, and to the first author's previous work concerning series by Wallis, Forsyth, and Ramanujan [23]; also, see [27]. Catalan and Fabry use a method due to Kummer [22] to accelerate the convergence of some series. Repeated application of Kummer's transformation leads to new series. One of the examples given by Fabry is the series

$$\frac{\pi^2}{12} = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^2}.$$
 (2)

This example was later taken over by Knopp where we find the following exercise [20, 129.c), p. 272]: Prove that

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^2(n+1)^2 \dots (n+p-1)^2} = \frac{5p+2}{4(p+1)p!^2} - \frac{p(p+1)^3}{4} \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^2(n+1)^2 \dots (n+p+1)^2}.$$

Using this recurrence with p = 1 transforms the series (2) into a new series that converges faster. Repeated application of the recurrence leads to this series:

$$\frac{\pi^2}{12} = \frac{7}{8} - \frac{8}{4} \cdot \left(\frac{17}{576} - \frac{192}{4} \cdot \left(\frac{27}{345600} - \frac{1080}{4} \cdot (\ldots)\right)\right) = \frac{1}{4} \sum_{n=1}^{\infty} (-1)^{n-1} \frac{10n - 3}{n^2 (2n - 1) \binom{2n}{n}}$$

with convergence rate $\frac{1}{4}$ [12, p.138]. The above recurrence can be rewritten and generalized in terms of hypergeometric functions. The result is given in the second part of Theorem 4.

While our hypergeometric recurrences as in Theorem 6 are of interest in their own right, the main results of this article are our new and fundamentally different proofs of known identities equating fundamental constants with fast convergent series, together with related evaluations for new fast converging series. We provide, for example, a new proof of Guillera's remarkable formula

$$\frac{8}{\pi^2} = \sum_{n=0}^{\infty} (-1)^n \left(\frac{1}{2}\right)^{12n} {2n \choose n}^5 (20n^2 + 8n + 1).$$
 (3)

that was introduced in 2003 in [14], in a completely different way compared to our methods, and we introduce generalizations of (3). It is remarkable how the WZ proof certificates for many of our recurrences are hugely complicated compared to the elegance and simplicity of our closed forms obtained from these recurrences, as in the Ramanujan-like formula in (3). Note however that the WZ proofs are straightforward compared to Fabry/Kummer-like proofs of these results. These require some ingenuity and are all based on special identities enabling a telescoping effect. For the sake of completeness, these identities are given at the end of this paper.

1.1 Related literature

Much of this article is inspired by the methods and results introduced in [18], in which the following fast convergent series for Catalan's constant $G = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$ were introduced:

$$G = \frac{1}{8} \sum_{n=1}^{\infty} \frac{16^n (6n-1)}{n^2 (2n-1) \binom{4n}{2n} \binom{2n}{n}},\tag{4}$$

$$G = \frac{1}{64} \sum_{n=1}^{\infty} \frac{(-1)^{n-1} 256^n (40n^2 - 24n + 3)}{n^3 (2n-1) \binom{4n}{2n}^2 \binom{2n}{n}},$$
(5)

$$G = \frac{1}{64} \sum_{n=1}^{\infty} \frac{256^n (580n^2 - 184n + 15)}{n^3 (2n-1) \binom{6n}{3n} \binom{6n}{4n} \binom{4n}{2n}}.$$
 (6)

The formula shown in (5) had actually been introduced and proved via the WZ method in 2000 by Lupaş in [25], prior to the 2010 article [18], and a completely different WZ-based proof, relative to [18, 25], of a formula equivalent to (5) was recently introduced in [6]. Lupaş' formula, as in (5), is used in the Wolfram Language to calculate G [26], which motivates our further exploring the use of WZ-based methods to prove and generalize evaluations for fast convergent series as in (5).

The series for G given by Lupaş in 2000 [25] and by Kh. Hessami Pilehrood and T. Hessami Pilehrood [18] and shown in (5) may be proved and generalized in a direct way using a new WZ-based method given in this article. Our alternate WZ-based proof of this G-formula is of interest in its own right, since it is completely different compared to the proof approaches used in [6, 18, 25]. The formula in (5) is highlighted as part of **Theorem 3** in [18], which gives a series acceleration for $\sum_{n=0}^{\infty} \beta(2n+2)a^{2n}$ for a complex number with |a| < 1, letting

$$\beta(s) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)^s}$$

denote Dirichlet's beta function. **Theorem 3** in [18] is proved using the Markov-WZ pair (F, G), where

$$F(n,k) = \frac{(-1)^{n+k} \left(\frac{1+a}{2}\right)_k \left(\frac{1-a}{2}\right)_k \left(1+\frac{a}{2}\right)_n \left(1-\frac{a}{2}\right)_n (2n)!}{\left(\frac{1-a}{2}\right)_{2n+k+1} \left(\frac{1+a}{2}\right)_{2n+k+1}}$$

and

$$G(n,k) = \frac{(-1)^{n+k} \left(\frac{1+a}{2}\right)_k \left(\frac{1-a}{2}\right)_k \left(1+\frac{a}{2}\right)_n \left(1-\frac{a}{2}\right)_n (2n)!}{8 \left(\frac{1-a}{2}\right)_{2n+k+2} \left(\frac{1+a}{2}\right)_{2n+k+2}} \cdot (40n^2 + 56n + 19 - a^2 + 24nk + 16k + 4k^2),$$

in contrast to our application of the WZ method, as opposed to the Markov-WZ method, to obtain the WZ proof certificate shown in our proof of Lemma 1. Lupaş' formula for G, as in [25], was proved via the WZ method in [25], and in a completely different way compared to our proof, with reference to the proof of **Theorem 3** in [25].

The above formulas for G also recall Guillera's [15] WZ-derived formula

$$\frac{\pi^2}{2} = \sum_{n=1}^{\infty} \frac{2^{4n} (3n-1)}{n^3 \binom{2n}{n}^3} \tag{7}$$

proved in [18] via the MWZ method, but already known to Catalan [7, (32), p. 18]. As in [6], we record that (5) was included in Lima's paper [24], where hypergeometric manipulations were used to prove a series expansion for Catalan's constant equivalent to the formula

$$\sum_{n=1}^{\infty} \frac{(-1)^n 2^{3n} (3n-1)}{n^3 \binom{2n}{n}^3} = -2G$$

proved by Guillera in [15].

Our series recursions and accelerations are also related to the work of Chu et al. in [8] and [10]. In [10], the series rearrangement referred to as the modified Abel lemma on summation by parts is employed to obtain hypergeometric series recursions related to our work. For a hypergeometric expression h(x, y) involving the variables x and y in its parameters, we prove recursions such as

$$h(x,y) = r_1(x,y) + r_2(x,y) h(x,y+1)$$

for rational functions $r_1(x, y)$ and $r_2(x, y)$, and we then apply such recursions iteratively, yielding accelerated series formulas. This same kind of approach is used in [10], but we do not make use of any Abel-type summation results, and we instead employ the WZ method, and, furthermore, our experimental approaches allow us to obtain recursions for ${}_{p}F_{q}(-1)$ -families, but such series do not seem to be considered in [10]. An advantage of our WZ-based methodologies is due to how the WZ difference equation

$$F(n+1,k) - F(n,k) = G(n,k+1) - G(n,k)$$

may manipulated in order to obtain new and nontrivial hypergeometric identities, using the WZ pairs introduced in this article, together with WZ-based series manipulations as in [5]. It appears that the Maple implementation of the WZ method cannot be applied to the recursions due to Chu and Zhang in [10], which shows how the methods in [10] are nontrivially different from ours, which rely on the WZ method.

1.2 Organization of the article

The rest of our article, apart from our concluding section on "non-computer" proofs, is mainly structured and organized in the manner indicated as follows: In each of the following main sections,

we introduce a recursion or recursions for families of ${}_{p}F_{q}(x)$ -series for fixed p, q, and x, and with the recurrences being given in terms of the parameters of the generalized hypergeometric functions under consideration. We present WZ proofs of these recursions, and then apply these recursions to obtain series acceleration identities, to obtain fast convergent series formulas for fundamental constants. Given one such fast convergent series, our method naturally gives rise to an infinite family, by altering the input parameters. In this regard, we choose to emphasize, as in Sections 3.1 and 5.1 below, our infinite families of generalizations of especially remarkable results due to Lupaş [25] and Guillera [14], recalling (3).

In Section 2, we introduce experimentally discovered $_3F_2(1)$ -recurrences, yielding formulas for very fast convergent series, such as

$$\zeta(2) = \frac{7}{4} - 2\sum_{n=2}^{\infty} \frac{(3n-1)(7n-5)}{n^2(n-1)^2 \binom{2n}{n}^3},\tag{8}$$

a series with convergence rate $\frac{1}{64}$.

In Section 3, we introduce experimentally discovered $_3F_2(-1)$ -recurrences, yielding an infinite family of series generalizing (5).

In Section 4, we introduce an experimentally discovered ${}_5F_4(-1)$ -recurrence, yielding new WZ proofs of two of Ramanujan's series for $\frac{1}{\pi}$, along with the new formula for Catalan's constant shown below, which does not follow from the G-evaluations highlighted in (4)–(6) or from the methods from past references as in [6, 18, 25] on series acceleration identities applied to Catalan's constant:

$$G = \frac{1}{16} \sum_{n=1}^{\infty} \frac{2^{6n} (336n^4 - 464n^3 + 216n^2 - 42n + 3)}{n^3 (2n-1)^3 \binom{4n}{2n}^3}.$$
 (9)

In Section 5, we introduce experimentally discovered $_6F_5(1)$ -recurrences, which we apply to obtain a variety of results, including a new proof of and generalizations of Guillera's remarkable formula (3) introduced in 2003 [14].

Finally, in Section 6, we provide "non-computer" proofs of our ${}_{p}F_{q}$ -recurrences, without using the WZ method.

2 Experimentally discovered $_3F_2(1)$ -recurrences

For the sake of brevity, we assume familiarity with WZ theory [28], generalized hypergeometric series, the Pochhammer symbol, etc., and we proceed with our first main result.

Theorem 1. The $_3F_2(1)$ -transform

$${}_{3}F_{2}\begin{bmatrix} x, x+1, 1 \\ x+y, x+y+1 \end{bmatrix} = \frac{2x+3y-1}{2(2y-1)} + \frac{(y-1)y(y+1)}{2(2y-1)(x+y)(x+y+1)} \cdot {}_{3}F_{2}\begin{bmatrix} x, x+1, 1 \\ x+y+1, x+y+2 \end{bmatrix}$$

holds true for x and y such that both sides of the above equality converge.

Proof. We set x = -n. So, it remains to prove that

$$\sum_{k=0}^{\infty} \frac{(1-n)_k(-n)_k}{(y-n)_k(-n+y+1)_k} \cdot \frac{2(2y-1)}{-2n+3y-1} \cdot \frac{2y^2(4k-4n+1)+y(4(k-n)^2-1)-2(k-n)(k-n+1)+3y^3}{2(2y-1)(k-n+y)(k-n+y+1)}$$

constantly equals 1. Now, we restrict n to being a member of the set \mathbb{N}_0 . In this case, Pochhammer symbols of the form $(-n)_k$ vanish for k > n, with $k \in \mathbb{N}_0$. So, for $n \in \mathbb{N}_0$, we may replace the upper limit of the above series with n. Now, we may apply the WZ method, giving us the following WZ proof certificate.

 $-(k-n+y)*(1+k-n+y)*(8*k^2*n^2*y-16*k^2*n*y^2+6*k^2*y^3-16*k*n^3*y+48*k*n^2*y^2-42*k*n*y^3+9*k*y^4+8*n^4*y-32*n^3*y^2+42*n^2*y^3-18*n*y^4-4*k^2*n^2+16*k^2*n*y-12*k^2*y^2+8*k*n^3-48*k*n^2*y+72*k*n*y^2-27*k*y^3-4*n^4+32*n^3*y-72*n^2*y^2+54*n*y^3-9*y^4-4*k^2*n+6*k^2*y+12*k*n^2-38*k*n*y+27*k*y^2-8*n^3+38*n^2*y-54*n*y^2+21*y^3+4*k*n-9*k*y-4*n^2+18*n*y-15*y^2+3*y)*k/(3+2*n-3*y)/(n+1-y)/(n-y)/(k-n)/(-n+k-1)/(4*k^2*y-8*k*n*y+8*k*y^2+4*n^2*y-8*n*y^2+3*y^3-2*k^2+4*k*n-2*n^2+2*y^2-2*k+2*n-y)$

So, using the WZ method, we have shown that the above infinite series identity holds for $n \in \mathbb{N}_0$. We may thus apply Carlson's theorem [3, p. 39] as in [11], to show that the same series identity holds for real n.

Theorem 2. The $_3F_2(1)$ -transform

$$_{3}F_{2}\begin{bmatrix} x & x+1 & 1 \\ x+y & x+y+1 \end{bmatrix} = 1 + \frac{x(x+1)}{(x+y)(x+y+1)} \cdot {}_{3}F_{2}\begin{bmatrix} x+1 & x+2 & 1 \\ x+y+1 & x+y+2 \end{bmatrix}$$

holds true for x and y such that both sides of the above equality converge.

Proof. Again, we set x = -n. So, it remains to prove that

$$\sum_{k=0}^{\infty} \frac{(1-n)_k(-n)_k}{(y-n)_k(-n+y+1)_k} \frac{y(2k-2n+y+1)}{(k-n+y)(k-n+y+1)} = 1.$$

Again, we consider the case for $n \in \mathbb{N}_0$, so that we may again replace the upper parameter of the above sum with n. Again, we may apply the WZ method; the corresponding WZ proof certificate is given below:

$$\frac{k(k-n+y)(k-n+y+1)(-2kn+ky-k+2n^2-2ny+2n-y+1)}{(k-n-1)(k-n)(n-y)(n-y+1)(2k-2n+y+1)}.$$

We may thus apply Carlson's theorem, as before.

2.1 A series acceleration method

Let us define

$$s(x,y) = \frac{x}{x+y} \cdot {}_{3}F_{2} \left[\begin{array}{c} x & x+1 & 1 \\ x+y & x+y+1 \end{array}; 1 \right] = \sum_{n=0}^{\infty} \frac{(x)_{n}(x)_{n+1}}{(x+y)_{n}(x+y)_{n+1}} .$$

We rewrite the recurrences given as Theorems 1 and 2, respectively, so that

$$s(x,y) = \frac{x(2x+3y-1)}{2(2y-1)(x+y)} + \frac{(y-1)y(y+1)}{2(2y-1)(x+y)^2}s(x,y+1)$$
(10)

and that

$$s(x,y) = \frac{x}{x+y} + \frac{x^2}{(x+y)^2} s(x+1,y). \tag{11}$$

Note that we know the value of s(x, y) in some cases:

$$s(1,2) = 7 - \frac{2}{3}\pi^2, s(\frac{1}{2},2) = 3 - \frac{9}{32}\pi^2, s(1,\frac{3}{2}) = 10 - 3\pi, s(\frac{1}{2},\frac{3}{2}) = 2 - \frac{16}{3\pi}.$$
 (12)

Indeed, we have, for instance, that:

$$s(1,2) = \sum_{n=0}^{\infty} \frac{4}{(n+3)(n+2)^2(n+1)} = \sum_{n=0}^{\infty} \left(\frac{2}{n+1} - \frac{2}{n+3} - \frac{4}{(n+2)^2} \right),$$

leading to the given closed form for s(1,2). We can deal with the second one in a similar manner. For the third and fourth ones, we can use a closed form for the partial sum of the series, as in:

$$s(\frac{1}{2}, \frac{3}{2}) = \sum_{n=0}^{\infty} \frac{2}{(n+1)(2n-1)} \frac{\binom{2n}{n}^2}{2^{4n}} \quad \text{with} \quad \sum_{n=1}^{k-1} \frac{2}{(n+1)(2n-1)} \frac{\binom{2n}{n}^2}{2^{4n}} = 2 - \frac{8}{3} \frac{k(4k-1)}{2k-1} \frac{\binom{2k}{k}^2}{2^{4k}}.$$

We then take the limit and use Wallis's product formula for π .

We can use the s-recurrences above to accelerate the convergence of the corresponding series. Iterating the first recurrence leads to the following accelerated series:

$$s(x,y) = \sum_{n=0}^{\infty} \frac{x(2x+3(y+n)-1)}{2(2(y+n)-1)(x+y+n)} \frac{(y-1)_n y_n (y+1)_n}{2^{2n} (y-\frac{1}{2})_n (x+y)_n^2}.$$

For the four special values given above, we find the following four series:

$$\zeta(2) = \frac{7}{4} - \sum_{n=2}^{\infty} \frac{3n+1}{(n+1)n^2(n-1)\binom{2n}{n}},$$

$$\zeta(2) = \frac{16}{9} - \sum_{n=2}^{\infty} \frac{2^{4n}}{(2n+1)n(n-1)\binom{2n}{n}^3},$$

$$\pi = -2\sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{4n} \binom{2n}{n} \frac{6n+5}{(2n+3)(2n+1)(2n-1)},$$

$$\frac{1}{\pi} = -12\sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{8n} \binom{2n}{n}^3 \frac{n^2}{(2n-1)(2n-3)}.$$
(13)

The fast convergent series evaluation in (13) meaningfully relates to the formulas

$$\frac{\pi^2}{4} = \sum_{n=1}^{\infty} \frac{2^{4n}(n+1)(3n+1)}{n(2n+1)^2 \binom{2n}{n}^3}$$
 (14)

and

$$2 + \frac{\pi^2}{2} = \sum_{n=1}^{\infty} \frac{2^{4n} (6n^2 + 2n - 1)}{n^2 (2n+1) \binom{2n}{n}^3}$$
 (15)

recently introduced in [6] and proved in [6] using WZ-based telescoping arguments together with WZ pairs obtained from Catalan number sum identities given in [9], noting that (13), (14), and (15) all have the same convergence rate of $\frac{1}{4}$.

If we use (11) in (10) we get a recurrence relating s(x,y) to s(x+1,y+1), and other accelerated sums, including (8). This series is especially remarkable for many reasons, perhaps most notably due to its very fast convergence rate of $\frac{1}{64}$. In the same manner that we have obtained (8), the following formulas may be proved.

$$\zeta(2) = \frac{16}{9} - 2\sum_{n=1}^{\infty} \frac{2^{4n}(14n^2 + 11n + 1)(n+1)}{n(4n+3)(4n+1)^2(2n-1)\binom{2n}{n}\binom{4n}{2n}^2}$$

$$\pi = \frac{10}{3} - 2\sum_{n=1}^{\infty} \frac{(14n+11)(2n+1)(3n+1)\binom{2n}{n}}{(4n+3)(4n+1)^2(2n-1)\binom{4n}{2n}^2}$$

$$\frac{1}{\pi} = -\frac{1}{8}\sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{12n} \binom{2n}{n}^3 \frac{(6n+1)(14n-3)(2n+1)}{(2n-1)^2}$$

This leads to an infinite number of possibilities.

3 Experimentally discovered $_3F_2(-1)$ -recurrences

In view of the closed forms shown in (12), it is not clear as to how to obtain accelerated series for Catalan's constant $G = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$. This motivates the exploration of alternating variants and generalizations of the above Lemmas. Defining t(x, y) so that

$$t(x,y) = \sum_{n=0}^{\infty} \frac{(x)_n (x)_{n+1}}{(x+y)_n (x+y)_{n+1}} (-1)^n,$$
(16)

we find that $t(\frac{1}{2},2) = \frac{3}{8}(6G-5)$. Note that

$$t(x,y) = \frac{x}{x+y} \cdot {}_{3}F_{2} \begin{bmatrix} x & x+1 & 1 \\ x+y & x+y+1 \end{bmatrix}; -1$$
.

This motivates the use of analogues of our above acceleration methods, in order to obtain accelerated series for G. In view of the resemblance of the formula (13) compared to the WZ-derived results in (14) and (15) from [6], this leads us to consider how recurrences for (16) may be applied to obtain analogues of the accelerated G-series

$$-16G - 8 = \sum_{n=1}^{\infty} \frac{(-1)^n 2^{8n} \left(40n^2 + 4n - 1\right)}{n^2 (4n+1) \binom{2n}{n} \binom{4n}{2n}^2}$$
(17)

proved in [6], which is equivalent to (5).

The recurrence highlighted below is remarkable in the sense that it is far from clear as to how our proof of Theorem 1 could be mimicked or reformulated so as to prove the identity highlighted as Lemma 1. We had discovered this recurrence experimentally.

Lemma 1. The recurrence

$$t(x,y) = \frac{x(2x^2 + 6xy + 5y^2 + 2x + 4y)}{4(x+y)^2(x+y+1)} - \frac{y^2(y+1)(y+2)}{4(x+y)^2(x+y+1)^2}t(x,y+2)$$
(18)

holds true for suitably bounded x and y.

Proof. Rewrite x as -n. Writing F(n,k) in place of the product of

$$\frac{(-1)^k(-n)_k(-n)_{k+1}}{(-n+y)_k(-n+y)_{1+k}}$$

and

$$-\frac{(n-y)^2(-n+y+1)\left(\frac{(y+1)(y+2)y^2}{(k-n+y)(k-n+y+1)^2(k-n+y+2)}+4\right)}{n\left(-6ny+2(n-1)n+5y^2+4y\right)},$$

the desired recurrence is equivalent to the identity

$$\sum_{k=0}^{\infty} F(n,k) = 1.$$

For the time being, we restrict our attention to the case whereby $n \in \mathbb{N}_0$. By the vanishing of $(-n)_k$ for $k > n \in \mathbb{N}_0$, with $k \in \mathbb{N}_0$, it remains to prove that

$$\sum_{k=0}^{n} F(n,k) = 1, \tag{19}$$

again for the $n \in \mathbb{N}_0$ case. We have successfully applied the WZ method to prove the above finite sum identity, and the corresponding WZ proof certificate is reproduced below.

 $-(1+k-n+y)*(2+k-n+y)*y*k*(8*k^3*n^3-28*k^3*n^2*y+32*k^3*n*y^2-10*k^3*y^3-24*k^2*n^4+112*k^2*n^3*y-192*k^2*n^2*y^2+136*k^2*n*y^3-30*k^2*y^4+24*k*n^5-140*k*n^4*y+320*k*n^3*y^2-348*k*n^2*y^3+170*k*n*y^4-25*k*y^5-8*n^6+56*n^5*y-160*n^4*y^2+232*n^3*y^3-170*n^2*y^4+50*n*y^5-8*k^3*n^2+20*k^3*n*y-6*k^3*y^2+32*k^2*n^3-112*k^2*n^2*y+104*k^2*n*y^2-18*k^2*y^3-40*k*n^4+184*k*n^3*y-276*k*n^2*y^2+142*k*n*y^3-5*k*y^4+16*n^5-92*n^4*y+184*n^3*y^2-142*n^2*y^3+10*n*y^4+25*y^5+4*k^3*y-8*k^2*n^2+12*k^2*y^2+16*k*n^3-24*k*n^2*y-12*k*n*y^2+26*k*y^3-8*n^4+16*n^3*y+12*n^2*y^2-52*n*y^3+35*y^4-4*k*n*y+2*k*y^2+4*n^2*y-4*n*y^2+2*y^3-4*k*y+8*n*y-8*y^2)/(n-y)/(2*n^2-6*n*y+5*y^2+2*n-2*y)/(-n+k-1)/(k-n)/(-1+n-y)/(4*k^4-16*k^3*n+16*k^3*y+24*k^2*n^2-48*k^2*n*y+24*k^2*y^2-16*n*y^3+5*y^4+16*k^3-48*k^2*y+48*k*n^2-96*k*n*y+48*k*y^2-16*n^3+48*n^2*y-48*n*y^2+19*y^3+20*k^2-40*k*n+40*k*y+20*n^2-40*n*y+22*y^2+8*k-8*n+8*y)$

So, we have shown that (19) holds for $n \in \mathbb{N}_0$. Again by the Pochhammer vanishing property referred to above, we have that

$$\sum_{k=0}^{\infty} F(n,k) = 1 \tag{20}$$

for $n \in \mathbb{N}_0$. Again, we may apply Carlson's theorem, in much the same way as in [11], to conclude that (20) also holds for real n.

Theorem 3. The expression t(x,y) equals the accelerated infinite series given as follows. Letting this series be over indices $j \in \mathbb{N}_0$, its summand may be written as the quotient of

$$(-1)^{j}4^{-j}(y)_{2j}\left(\frac{y}{2}\right)_{j}\left(\frac{y+2}{2}\right)_{j}x\left(2x+2x^{2}+4(2j+y)+6x(2j+y)+5(2j+y)^{2}\right)$$

by

$$4(x+y)_{2j+1}^{2}(x+y+2j+1)$$

Proof. Through repeated applications of the recurrence given as Lemma 1, we may rewrite t(x,y) as

$$\sum_{j=0}^{m-1} \frac{(-1)^j \left(\prod_{i=0}^{j-1} \frac{(2i+y)^2(1+2i+y)(2+2i+y)}{4(2i+x+y)^2(1+2i+x+y)^2}\right) x \left(2x+2x^2+4(2j+y)+6x(2j+y)+5(2j+y)^2\right)}{4(2j+x+y)^2(1+2j+x+y)} + \frac{(-1)^j \left(\prod_{i=0}^{j-1} \frac{(2i+y)^2(1+2i+y)(2+2i+y)}{4(2i+x+y)^2(1+2i+x+y)}\right) x \left(2x+2x^2+4(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+6x(2j+y)+$$

$$(-1)^m \left(\prod_{i=0}^{m-1} \frac{(2i+y)^2(1+2i+y)(2+2i+y)}{4(2i+x+y)^2(2i+x+y+1)^2} \right) t(x,y+2m).$$

Taking $m \to \infty$, it is easily seen that the latter term vanishes.

Example 1. Setting $x = \frac{1}{2}$ and y = 1 in Theorem 3, we obtain the accelerated series identity

$$8 - 4\pi = \sum_{j=1}^{\infty} \frac{(-1)^j 2^{4j} \left(40j^2 - 12j - 1\right) {2j \choose j}}{j(2j-1)^2 (4j+1) {4j \choose 2j}^2}.$$

The above series is of convergence rate $\frac{1}{4}$.

3.1A family of accelerated series for G

Setting $x = -\frac{1}{2}$ and y = 2 in Theorem 3 gives us the series (17). By setting $x = \frac{1}{2}$ and y = 2 in Theorem 3, we may obtain an alternate proof of the formula (5) from [18]. Explicitly, we obtain that

$$\frac{40}{3} - 16G = \sum_{j=1}^{\infty} \frac{(-1)^j 2^{8j} \left(40j^2 + 28j + 3\right)}{j^2 (4j+1)^2 (4j+3) \binom{2j}{j} \binom{4j}{2j}^2},\tag{21}$$

and adding 16 times the series in (5) gives us a series with partial sums that are evaluable in closed form. However, it is not clear as to how to obtain the G-series given below using Markov-WZ methood together with series involving Dirichlet's β function, as in [18].

Example 2. Set $x = \frac{3}{2}$ and y = 2 in Theorem 3. This gives us

$$\frac{16G}{3} - \frac{664}{135} = \sum_{j=1}^{\infty} \frac{(-1)^j 2^{8j} (40j^2 + 52j + 15)}{j^2 (4j+1)^2 (4j+3)^2 (4j+5) \binom{2j}{i} \binom{4j}{2i}^2}.$$

Example 3. Set $x = \frac{5}{2}$ and y = 2 in Theorem 3. This gives us that

$$\frac{23048}{70875} - \frac{16G}{45} = \sum_{j=0}^{\infty} \frac{(-1)^j 2^{8j} (40j^2 + 76j + 35)}{j^2 (4j+1)^2 (4j+3)^2 (4j+5)^2 (4j+7) \binom{2j}{j} \binom{4j}{2j}^2}.$$

Continuing in this manner gives us an infinite family of extensions of (21). It appears that our techniques for generating such families of accelerated series have not been considered previously, as in past literature related to [18], including [1, 2, 16, 19, 21, 30].

3.2 Further ${}_{3}F_{2}(-1)$ -recurrences

Theorem 4. The recurrences

$${}_{3}F_{2}\begin{bmatrix} x, x, 1 \\ x + y, x + y \end{bmatrix}; -1 = 1 - \frac{x^{2}}{(x+y)^{2}} \cdot {}_{3}F_{2}\begin{bmatrix} x + 1, x + 1, 1 \\ x + 1 + y, x + 1 + y \end{bmatrix}; -1$$

$${}_{3}F_{2}\begin{bmatrix} x, x, 1 \\ x + y, x + y \end{bmatrix}; -1 = \frac{2x^{2} + 6xy + 5y^{2} + y}{4(x+y)^{2}} - \frac{y(y+1)^{3}}{4(x+y)^{2}(x+y+1)^{2}} \cdot {}_{3}F_{2}\begin{bmatrix} x, x, 1 \\ x + y + 2, x + y + 2 \end{bmatrix}; -1$$

hold true, if all of the above series converge.

Proof. Write x = -n. We may mimic our previous WZ proofs, via Carlson's theorem, to prove the first out of the two ${}_{3}F_{2}(-1)$ -recurrences shown above. a Corresponding WZ certificate is shown below:

$$\frac{ky(k-n+y)^2(-2kn+ky-2k+2n^2-2ny+4n-2y+2)}{(k-n-1)^2(n-y+1)^2(2k^2-4kn+2ky+2n^2-2ny+y^2)}.$$

We may apply the same approach to prove the latter ${}_{3}F_{2}(-1)$ -formula under consideration; a WZ proof certificate is given below.

 $-(8*k^3*n^3-28*k^3*n^2*y+32*k^3*n*y^2-10*k^3*y^3-24*k^2*n^4+112*k^2*n^3*y-192*k^2*n^2*y^2+136*k^2*n*y^3-30*k^2*y^4+24*k*n^5-140*k*n^4*y+320*k*n^3*y^2-348*k*n^2*y^3+170*k*n*y^4-25*k*y^5-8*n^6+56*n^5*y-160*n^4*y^2+232*n^3*y^3-170*n^2*y^4+50*n*y^5+4*k^3*n^2-8*k^3*n*y+10*k^3*y^2-16*k^2*n^3+56*k^2*n^2*y-88*k^2*n*y^2+50*k^2*y^3+20*k*n^4-96*k*n^3*y+204*k*n^2*y^2-206*k*n*y^3+80*k*y^4-8*n^5+48*n^4*y-136*n^3*y^2+206*n^2*y^3-160*n*y^4+50*y^5+6*k^3*y-24*k^2*n*y+18*k^2*y^2+36*k*n^2*y-54*k*n*y^2+20*k*y^3-24*n^3*y+54*n^2*y^2-40*n*y^3+10*y^4+2*k^3-8*k^2*n-2*k^2*y+12*k*n^2+6*k*n*y-14*k*y^2-8*n^3-6*n^2*y+28*n*y^2-14*y^3-4*k^2+12*k*n-7*k*y-12*n^2+14*n*y-2*y^2+2*k-4*n+4*y)*k*y*(1+k-n+y)^2/(2*n^2-6*n*y+5*y^2+4*n-5*y+2)/(4*k^4-16*k^3*n+16*k^3*y+24*k^2*n^2-48*k^2*n*y+24*k^2*y^2-16*n*y^3+5*y^4+8*k^3-24*k^2*n+24*k^2*y+24*k^2*y^2-16*n*y^3+5*y^4+8*k^3-24*k^2*n+24*k^2*y+24*k^2*y+24*k*n^2+2*y^2+16*k*y^3+4*n^4-16*n^3*y+24*n^2*y^2-16*n*y^3+5*y^4+8*k^3-24*k^2*n+24*k^2*y+24*k*n^2-48*k*n*y+24*k*y^2-8*n^3+24*n^2*y-24*n*y^2+11*y^3+4*k^2-8*k*n+8*k*y+4*n^2-8*n*y+7*y^2+y)/(n-y)^2/(-n+k-1)^2$

With the notation

$$s(x,y) = {}_{3}F_{2} \begin{bmatrix} x & x & 1 \\ x+y & x+y \end{bmatrix},$$

we may express the recurrences in Theorem 4 as follows:

$$s(x,y) = 1 - \frac{x^2}{(x+y)^2} s(x+1,y),$$

$$s(x,y) = \frac{2x^2 + 6xy + 5y^2 + y}{4(x+y)^2} - \frac{y(y+1)^3}{4(x+y)^2(x+y+1)^2} s(x,y+2).$$

If we use the second recurrence to calculate $s(\frac{1}{2},2) = \frac{9}{2}(1-\frac{\pi}{4})$, $s(1,2) = 4(3-4\ln(2))$, $s(\frac{1}{2},1) = G$, we find the corresponding series in Lupaş [25, p. 6]. If we combine the recurrences to write s(x,y) in function of s(x+1,y+2) and of s(x+2,y+2), we get for $x=\frac{1}{2}$ and y=1 the series for Catalan's constant found in [18, pp. 233–234].

4 Experimentally discovered ${}_{5}F_{4}(-1)$ -recurrences

Theorem 5. The recurrences

$$\left(x + \frac{y-1}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{c} x, x, x, x + \frac{y+1}{2}, 1 \\ x + y, x + y, x + y, x + \frac{y-1}{2}; -1 \end{array} \right] = x + \frac{y-1}{2}$$

$$- \frac{x^{3}}{(x+y)^{3}} \left(x + 1 + \frac{y-1}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{c} x + 1, x + 1, x + 1, x + 1 + \frac{y+1}{2}, 1 \\ x + 1 + y, x + 1 + y, x + 1 + y, x + 1 + \frac{y-1}{2}; -1 \right]$$

$$\left(x + \frac{y-1}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{c} x, x, x, x + \frac{y+1}{2}, 1 \\ x + y, x + y, x + y, x + \frac{y-1}{2}; -1 \end{array} \right] = \frac{x-1}{2} + y$$

$$- \frac{y^{3}}{(x+y)^{3}} \left(x + \frac{y}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{c} x, x, x, x + \frac{y+2}{2}, 1 \\ x + y + 1, x + y + 1, x + y + 1, x + \frac{y}{2}; -1 \end{array} \right]$$

hold true, if all of the above series converge.

Proof. Again, we set x = -n, and we may apply the WZ method in much the same way as before. A WZ proof certificate for the first out of the two recursions shown above is provided as follows.

 $-(k-n+y)^3*(12*k^3*n^3*y-18*k^3*n^2*y^2+10*k^3*n*y^3-2*k^3*y^4-36*k^2*n^4*y+72*k^2*n^3*y^2-49*k^2*n^2*y^3+13*k^2*n*y^4-k^2*y^5+36*k*n^5*y-90*k*n^4*y^2+78*k*n^3*y^3-27*k*n^2*y^4+3*k*n*y^5-12*n^6*y+36*n^5*y^2-39*n^4*y^3+18*n^3*y^4-3*n^2*y^5-4*k^3*n^3+42*k^3*n^2*y-42*k^3*n*y^2+12*k^3*y^3+12*k^2*n^4-168*k^2*n^3*y+234*k^2*n^2*y^2-104*k^2*n*y^3+13*k^2*y^4-12*k*n^5+210*k*n^4*y-384*k*n^3*y^2+240*k*n^2*y^3-54*k*n*y^4+3*k*y^5+4*n^6-84*n^5*y+192*n^4*y^2-160*n^3*y^3+54*n^2*y^4-6*n*y^5-12*k^3*n^2+48*k^3*n*y-24*k^3*y^2+48*k^2*n^3-285*k^2*n^2*y+249*k^2*n*y^2-54*k^2*y^3-60*k*n^4+474*k*n^3*y-603*k*n^2*y^2+243*k*n*y^3-27*k*y^4+24*n^5-237*n^4*y+402*n^3*y^2-243*n^2*y^3+54*n*y^4-3*y^5-12*k^3*n+18*k^3*y+72*k^2*n^2-210*k^2*n*y+87*k^2*y^2-120*k*n^3+522*k*n^2*y-414*k*n*y^2+81*k*y^3+60*n^4-348*n^3*y+414*n^2*y^2-162*n*y^3+18*y^4-4*k^3+48*k^2*n-57*k^2*y-120*k*n^2+282*k*n*y-105*k*y^2+80*n^3-282*n^2*y+210*n*y^2-40*y^3+12*k^2-60*k*n+60*k*y+60*n^2-120*n*y+42*y^2-12*k+24*n-21*y+4)*k/(n+1-y)^3/(2*n+3-y)/(-n+k-1)^3/(4*k^4-16*k^3*n+8*k^3*y+24*k^2*n^2-24*k^2*n^2-24*k^2*n^2+9*k^2*y^2-16*k*n^3+24*k*n^2*y-18*k*n*y^2+5*k*y^3+4*n^4-8*n^3*y+9*n^2*y^2-5*n*y^3+y^4-3*k^2*y+6*k*n*y-3*k*y^2-3*n^2*y+3*n*y^2-y^3)$

A WZ proof certificate for the latter recursion is given below.

 $-k*(3*k^3*n^3*y-9*k^3*n^2*y^2+7*k^3*n*y^3-2*k^3*y^4-9*k^2*n^4*y+36*k^2*n^3*y^2-46*k^2*n^2*y^3+22*k^2*n*y^4-4*k^2*y^5+9*k*n^5*y-45*k*n^4*y^2+78*k*n^3*y^3-54*k*n^2*y^4+12*k*n*y^5-3*n^6*y+18*n^5*y^2-39*n^4*y^3+36*n^3*y^4-12*n^2*y^5-k^3*n^3+12*k^3*n^2*y-21*k^3*n*y^2+8*k^3*y^3+3*k^2*n^4-48*k^2*n^3*y+126*k^2*n^2*y^2-102*k^2*n*y^3+24*k^2*y^4-3*k*n^5+60*k*n^4*y-210*k*n^3*y^2+258*k*n^2*y^3-114*k*n*y^4+12*k*y^5+n^6-24*n^5*y+105*n^4*y^2-172*n^3*y^3+114*n^2*y^4-24*n*y^5-3*k^3*n^2+15*k^3*n*y-12*k^3*y^2+12*k^2*n^3-90*k^2*n^2*y+144*k^2*n*y^2-56*k^2*y^3-15*k*n^4+150*k*n^3*y-360*k*n^2*y^2+282*k*n*y^3-60*k*y^4+6*n^5-75*n^4*y+240*n^3*y^2-282*n^2*y^3+120*n*y^4-12*y^5-3*k^3*n+6*k^3*y+18*k^2*n^2-72*k^2*n*y+54*k^2*y^2-30*k*n^3+180*k*n^2*y-270*k*n*y^2+102*k*y^3+15*n^4-120*n^3*y+270*n^2*y^2-204*n*y^3+42*y^4-k^3+12*k^2*n-21*k^2*y-30*k*n^2+105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y^2+20*n^3-105*n^2*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y+150*n*y^2-55*y^3+3*k^2-15*k*n+24*k*y+15*n^2-105*k*n*y-75*k*y+15*n^2-105*k*n*y-75*k*y+15*n^2-105*k*n*y+15*n^2+105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y+15*n^2-105*k*n*y$

 $48*n*y+33*y^2-3*k+6*n-9*y+1)*(k-n+y)^3/(2*k^4-8*k^3*n+7*k^3*y+12*k^2*n^2-21*k^2*n*y+9*k^2*y^2-8*k*n^3+21*k*n^2*y-18*k*n*y^2+7*k*y^3+2*n^4-7*n^3*y+9*n^2*y^2-7*n*y^3+2*y^4-k^3+3*k^2*n-3*k^2*y-3*k*n^2+6*k*n*y-3*k*y^2+n^3-3*n^2*y+3*n*y^2-y^3)/(2+n-2*y)/(-n+k-1)^3/(n+1-y)^3$

With the notation:

$$s(x,y) = \left(x + \frac{y-1}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{cccc} x & x & x & x + \frac{y+1}{2} & 1 \\ x+y & x+y & x+y & x + \frac{y-1}{2} \end{array}; -1 \right],$$

we find that:

$$s(x,y) = x + \frac{y-1}{2} - \frac{x^3}{(x+y)^3} s(x+1,y),$$

$$s(x,y) = \frac{x-1}{2} + y - \frac{y^3}{(x+y)^3} s(x,y+1).$$

Note that for $x = \frac{1}{2}$, $y = \frac{1}{2}$, s(x, y) is Bauer's series (see f.i. [23]):

$$\frac{4}{\pi} = \sum_{n=0}^{\infty} (-1)^n \left(\frac{1}{2}\right)^{6n} {2n \choose n}^3 (4n+1)$$

If we try to accelerate this series with one of the recurrences, we get Bauer's series again in both cases. Hence no acceleration. But if we use them one after the other, we get Ramanujan's series for $\frac{1}{\pi}$:

$$\frac{16}{\pi} = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{12n} {2n \choose n}^3 (42n+5).$$

For $x = \frac{1}{2}$, y = 1, we have that

$$s(x,y) = \frac{1}{2} \sum_{n=0}^{\infty} (-1)^n \frac{1}{(2n+1)^2} = \frac{1}{2}G.$$

Using both recurrences one after the other leads to (9). For x = 1, y = 1, we obtain in this case:

$$\frac{1}{16} \sum_{n=0}^{\infty} \frac{21n+13}{(2n+1)^3 \binom{2n}{n}^3} = \frac{\pi^2}{12}.$$

(equivalent with [17, (7)]).

5 Experimentally discovered $_6F_5(1)$ -recurrences

Theorem 6. The recurrences

$$\left(x + \frac{y-1}{2}\right) \cdot {}_{6}F_{5} \left[\begin{array}{c} x, x, x, x, x + \frac{y+1}{2}, 1 \\ x + y, x + y, x + y, x + y, x + \frac{y-1}{2}; 1 \end{array} \right] = x + \frac{y-1}{2}$$

$$+ \frac{x^{4}}{(x+y)^{4}} \left(x + 1 + \frac{y-1}{2}\right) \cdot {}_{6}F_{5} \left[\begin{array}{c} x + 1, x + 1, x + 1, x + 1, x + 1 + \frac{y+1}{2}, 1 \\ x + 1 + y, x + 1 + \frac{y-1}{2}; 1 \right]$$

$$\left(x + \frac{y-1}{2}\right) \cdot {}_{6}F_{5} \left[\begin{array}{c} x, x, x, x, x + \frac{y+1}{2}, 1 \\ x + y, x + y, x + y, x + y, x + \frac{y-1}{2}; 1 \end{array} \right] = \frac{5y^{2} + 6(x-1)y + 2(x-1)^{2}}{4(2y-1)}$$

$$- \frac{y^{5}}{2(2y-1)(x+y)^{4}} \left(x + \frac{y}{2}\right) \cdot {}_{6}F_{5} \left[\begin{array}{c} x, x, x, x, x + \frac{y+2}{2}, 1 \\ x + y + 1, x + \frac{y}{2}; 1 \end{array} \right]$$

hold true, if all of the above series converge.

Proof. Again, we write x = -n, and we may again mimic our WZ-based approaches utilized as above. For the first $_6F_5(1)$ -recurrence on display in the above theorem, a corresponding WZ proof certificate is shown below.

 $-(-n+y+k)^4*(16*k^4*n^4*y-32*k^4*n^3*y^2+28*k^4*n^2*y^3-12*k^4*n*y^4+2*k^4*y^5-12*k^4*n^2*y^3-12*k^4*n^4*y^5-12*k^4*n^4*y^5-12*k^4*n^4*y^5-12*k^4*n^4*y^5-12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*n^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6+12*k^4*y^6$ $64*k^3*n^5*y+160*k^3*n^4*y^2-156*k^3*n^3*y^3+74*k^3*n^2*y^4-16*k^3*n*y^5+k^3*y^6+16*k^3*n^2*y^4-16*k^3*n^4*y^5+160*k^3*n^4*y^5+160*k^3*n^4*y^6+160*k^3*n^4*y^5+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*n^4*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y^6+160*k^3*y$ $96*k^2*n^6*y-288*k^2*n^5*y^2+328*k^2*n^4*y^3-176*k^2*n^3*y^4+44*k^2*n^2*y^5-176*k^2*n^6*y-288*k^2*n^6*y^2+328*k^2*n^4*y^3-176*k^2*n^3*y^4+44*k^2*n^2*y^5-176*k^2*n^4*y^3-176*k^2*n^4*y^3-176*k^2*n^4*y^4+44*k^2*n^4*y^5-176*k^2*n^4*y^4+44*k^2*n^4*y^3-176*k^2*n^4*y^4+44*k^2*n^4*y^3-176*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*k^2*n^4*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+44*y^4+4$ $4*k^2*n*y^6-64*k*n^7*y+224*k*n^6*y^2-300*k*n^5*y^3+190*k*n^4*y^4-56*k*n^3*y^5+190*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56*k*n^4*y^4-56$ $6*k*n^2*y^6+16*n^8*y-64*n^7*y^2+100*n^6*y^3-76*n^5*y^4+28*n^4*y^5-4*n^3*y^6-10*n^6*y^3-76*n^5*y^4+28*n^4*y^5-4*n^3*y^6-10*n^6*y^3-76*n^5*y^4+28*n^4*y^5-4*n^3*y^6-10*n^6*y^3-76*n^5*y^4+28*n^4*y^5-4*n^3*y^6-10*n^6*y^3-76*n^5*y^4+28*n^4*y^5-4*n^3*y^6-10*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*n^6*y^3-76*n^6*y^3-76*n^6*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^3-76*n^6*y^$ $360*k^3*n^4*y+688*k^3*n^3*y^2-500*k^3*n^2*y^3+156*k^3*n*y^4-16*k^3*y^5-24*k^2*n^6+12*k^3*n^4*y^5-12*k^2*n^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*n^4*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12*k^3*y^6+12$ 4*k^2*y^6+16*k*n^7-504*k*n^6*y+1416*k*n^5*y^2-1540*k*n^4*y^3+768*k*n^3*y^4- $168*k*n^2*y^5+12*k*n*y^6-4*n^8+144*n^7*y-472*n^6*y^2+616*n^5*y^3-384*n^4*y^4+168*k*n^2*y^5+12*k*n^2*y^6-4*n^8+144*n^7*y-472*n^6*y^2+616*n^5*y^3-384*n^4*y^4+168*k*n^2*y^6-4*n^8+144*n^7*y-472*n^6*y^2+616*n^5*y^3-384*n^4*y^4+168*n^6*y^6-4*n^8+144*n^7*y-472*n^6*y^2+616*n^5*y^3-384*n^4*y^4+168*n^6*y^6-4*n^8+144*n^7*y-472*n^6*y^6-4*n^8+144*n^7*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*y^6-472*n^6*$ $112*n^3*y^5-12*n^2*y^6-16*k^4*n^3+120*k^4*n^2*y-120*k^4*n*y^2+36*k^4*y^3+$ $80*k^3*n^4-796*k^3*n^3*y+1098*k^3*n^2*y^2-528*k^3*n*y^3+81*k^3*y^4-144*k^2*n^5+128*k^3*p^4-144*k^2+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+128*p^4+1$ 1788*k^2*n^4*y-3192*k^2*n^3*y^2+2096*k^2*n^2*y^3-548*k^2*n*y^4+44*k^2*y^5+ $112*k*n^6-1668*k*n^5*y+3690*k*n^4*y^2-3136*k*n^3*y^3+1158*k*n^2*y^4-168*k*n*y^5+$ $6*k*y^6-32*n^7+556*n^6*y-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^5*y^2+1568*n^4*y^3-772*n^3*y^4+168*n^2*y^5-1476*n^3*y^4+168*n^4*y^3-1476*n^4*y^5-1476*n^4*y^5-1476*n^4*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5*y^5-1476*n^5$ $12*n*y^6-24*k^4*n^2+88*k^4*n*y-44*k^4*y^2+160*k^3*n^3-868*k^3*n^2*y+772*k^3*n*y^2-12*n*y^6-24*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4*n^2+88*k^4$ $184*k^3*v^3-360*k^2*n^4+2592*k^2*n^3*y-3312*k^2*n^2*y^2+1424*k^2*n*y^3-184*k^2*n^2*y^2+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*n^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1424*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+1444*k^2*y^3+444*y^3+444*y^3+444*y^3+444*y$ 184*k^2*y^4+336*k*n^5-3020*k*n^4*y+5080*k*n^3*y^2-3168*k*n^2*y^3+772*k*n*y^4- $56*k*y^5-112*n^6+1208*n^5*y-2540*n^4*y^2+2112*n^3*y^3-772*n^2*y^4+112*n*y^5-4*y^6-1208*n^5*y^5-124*y^6-124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n^6+124*n$ $16*k^4*n+24*k^4*y+160*k^3*n^2-468*k^3*n*y+202*k^3*y^2-480*k^2*n^3+2088*k^2*n^2*y-16*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^3*n^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*k^2+160*$ $1704*k^2*n*y^2+360*k^2*y^3+560*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^2*y^2+360*k^2*y^3+560*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^3*y+3900*k*n^2*y^2-1588*k*n*y^3+360*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3240*k*n^4-3$ $192*k*y^4-224*n^5+1620*n^4*y-2600*n^3*y^2+1588*n^2*y^3-384*n*y^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*y^5-4*k^4+28*$ $1584*k*n*y^2-316*k*y^3-280*n^4+1376*n^3*y-1584*n^2*y^2+632*n*y^3-76*y^4+16*k^3-1584*n^2*y^2+632*n^2+16*k^3-1584*n^2+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+16*k^3+k^3+16*k^3+16*k^3+k^3+16*k^3+k^3+16*k^3+16*k^3+16*k^3+k^3+k^3+k^3+k^3+k^3+k^3+k^3+k^3+k$ 144*k^2*n+156*k^2*y+336*k*n^2-724*k*n*y+266*k*y^2-224*n^3+724*n^2*y-532*n*y^2+ $104*y^3-24*k^2+112*k*n-108*k*y-112*n^2+216*n*y-76*y^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+112*k*n-108*k*y-112*n^2+216*n*y-76*y^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+112*k*n-108*k*y-112*n^2+216*n*y-76*y^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+112*k*n-108*k*y-112*n^2+216*n*y-76*y^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+112*k*n-108*k*y-112*n^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+112*k^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^2+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+1-104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+104*y^3-24*k^3+16*k-32*n+28*y-4)*k/(n+104*y^3-24*k-28*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*y-4)*k/(n+104*y^3-24*k-24*y-4)*k/(n+104*y^3-24*y-4)*k/(n+104*y^3-24*y-4)*k/(n+104*y^3-24*y-4)*k/(n+104*y^3-24*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+104*y-4)*k/(n+1$ $y)^4/(2*n+3-y)/(-n+k-1)^4/(8*k^4*y-32*k^3*n*y+16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*n^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^3*y^2+48*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*y-16*k^2*$ 48*k^2*n*y^2+14*k^2*y^3-32*k*n^3*y+48*k*n^2*y^2-28*k*n*y^3+6*k*y^4+8*n^4*y $16*n^3*y^2+14*n^2*y^3-6*n*y^4+y^5-2*k^4+8*k^3*n-4*k^3*y-12*k^2*n^2+12*k^2*n*y-6*k^2*y^2+8*k*n^3-12*k*n^2*y+12*k*n*y^2-4*k*y^3-2*n^4+4*n^3*y-6*n^2*y^2+4*n*y^3-y^4)$

A corresponding proof certificate for the latter recursion is given below.

 $-(16*k^5*n^5*y-72*k^5*n^4*y^2+128*k^5*n^3*y^3-112*k^5*n^2*y^4+52*k^5*n*y^5-12*k^5*n^2*y^4+52*k^5*n^4*y^5-12*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^4*y^5+128*k^5*n^5*n^5*y^5+128*k^5*n^5*y^5+128*k^5*n^5*y^5+128*k^5*n^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+128*k^5*y^5+$ 10*k^5*y^6-80*k^4*n^6*y+432*k^4*n^5*y^2-944*k^4*n^4*y^3+1056*k^4*n^3*y^4- $648*k^4*n^2*y^5+216*k^4*n*y^6-30*k^4*y^7+160*k^3*n^7*y-1008*k^3*n^6*y^2+$ $2624*k^3*n^5*y^3-3600*k^3*n^4*y^4+2780*k^3*n^3*y^5-1210*k^3*n^2*y^6+280*k^3*n*y^7-1210*k^3*n^2*y^6+280*k^3*n^4*y^4+2780*k^3*n^3*y^5-1210*k^3*n^2*y^6+280*k^3*n^4*y^4+2780*k^3*n^3*y^5-1210*k^3*n^2*y^6+280*k^3*n^4*y^4+2780*k^3*n^3*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^3*n^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*k^4*y^6+280*$ 25*k^3*y^8-160*k^2*n^8*y+1152*k^2*n^7*y^2-3488*k^2*n^6*y^3+5696*k^2*n^5*y^4-5360*k^2*n^4*y^5+2880*k^2*n^3*y^6-820*k^2*n^2*y^7+100*k^2*n*y^8+80*k*n^9*y- $648*k*n^8*y^2+2240*k*n^7*y^3-4256*k*n^6*y^4+4764*k*n^5*y^5-3110*k*n^4*y^6+$ $1080*k*n^3*y^7-150*k*n^2*y^8-16*n^10*y+144*n^9*y^2-560*n^8*y^3+1216*n^7*y^4-1080*k*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+1216*n^2*y^3+n^2*y^3+n^2*y^3+$ $1588*n^6*y^5+1244*n^5*y^6-540*n^4*y^7+100*n^3*y^8-8*k^5*n^5+124*k^5*n^4*y-100*n^3*y^8-8*k^5*n^5+124*k^5*n^4*y-100*n^3*y^8-8*k^5*n^5+124*k^5*n^4*y-100*n^4*y^8-8*k^5*n^5+124*k^5*n^4*y-100*n^4*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*n^6*y^8-8*k^6*y^8-8*k^6*n^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8-8*k^6*y^8
384*k^5*n^3*y^2+488*k^5*n^2*y^3-280*k^5*n*y^4+64*k^5*y^5+40*k^4*n^6-744*k^4*n^5*y+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^5*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y^3+64*k^3*n^2*y$ 2848*k⁴*n⁴*y²-4688*k⁴*n³*y³+3816*k⁴*n²*y⁴-1536*k⁴*n*y⁵+252*k⁴*y⁶- $80*k^3*n^7+1736*k^3*n^6*y-7936*k^3*n^5*y^2+16120*k^3*n^4*y^3-17040*k^3*n^3*y^4+16120*k^3*n^4*y^3-17040*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^3*n^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+16120*k^4*y^4+161$ $9620*k^3*n^2*y^5-2740*k^3*n*y^6+310*k^3*y^7+80*k^2*n^8-1984*k^2*n^7*y+$ $10560*k^2*n^6*y^2-25600*k^2*n^5*y^3+33440*k^2*n^4*y^4-24400*k^2*n^3*y^5+10560*k^2*n^4*y^4-24400*k^2*n^4+10560*k^2*n^4*y^4-24400*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10560*k^2*n^4+10660*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4+1060*k^2*n^4$ $9560*k^2*n^2*y^6-1760*k^2*n*y^7+100*k^2*y^8-40*k*n^9+1116*k*n^8*y-6784*k*n^7*y^2+1116*k*n^8*y^8-40*k^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+1116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n^8+116*k*n$ 19152*k*n^6*y^3-29904*k*n^5*y^4+26980*k*n^4*y^5-13640*k*n^3*y^6+3420*k*n^2*y^7-
$300*k*n*y^8+8*n^10-248*n^9*y+1696*n^8*y^2-5472*n^7*y^3+9968*n^6*y^4-10792*n^5*y^5+1000*k*n*y^8+8*n^10-248*n^9*y+1696*n^8*y^2-5472*n^7*y^3+9968*n^6*y^4-10792*n^5*y^5+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000*k*n^8*y^8+1000$ $608*k^5*n*y^3-172*k^5*y^4+264*k^4*n^5-2628*k^4*n^4*y+7264*k^4*n^3*y^2-1264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7264*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+7464*k^4*n^4*y+746*k^4*n^4*y+746*k^4*n^4*y+746*k^4*n^4*y+746*k^4*n^4*y+746*k^4*n^4*y+746*k^4*y+746*$ $8568*k^4*n^2*y^3+4536*k^4*n*y^4-900*k^4*y^5-616*k^3*n^6+7344*k^3*n^5*y-616*k^3*n^6+7344*k^3*n^5*y-616*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+7344*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*k^3*n^6+744*n^6+744*k^3*n^6+744*k^3*n^6+74*k^3*n^6+74*k^6+74*k^6+74*k^6+74*k^6+74*k^6+74*k^6+74*k^$ 25160*k^3*n^4*y^2+38880*k^3*n^3*y^3-29880*k^3*n^2*y^4+11000*k^3*n*y^5- $1540*k^3*y^6+704*k^2*n^7-9784*k^2*n^6*y+40080*k^2*n^5*y^2-76840*k^2*n^4*y^3+1240*k^2*n^2*n^2+1240*k^2*n^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+1240*k^2+124$
$6288*k*n^7*y-30016*k*n^6*y^2+68880*k*n^5*y^3-86520*k*n^4*y^4+60640*k*n^3*y^5-128880*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60640*k*n^4*y^4+60660*k*n^4*y^4+60660*k*n^4*y^4+6060*k*n^4*y^4+6060*k*n^4*y^4+6060*k*n^4*y^4+6060*k*n^4*y^4+6060*k*n^4*y^4+606$ 22320*k*n^2*y^6+3600*k*n*y^7-150*k*y^8+88*n^9-1572*n^8*y+8576*n^7*y^2- $22960*n^6*y^3+34608*n^5*y^4-30320*n^4*y^5+14880*n^3*y^6-3600*n^2*y^7+300*n*y^8-14880*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3600*n^2*y^6-3$ $96*k^5*n^3+472*k^5*n^2*y-624*k^5*n*y^2+248*k^5*y^3+720*k^4*n^4-4672*k^4*n^3*y+1248*k^5*n^2+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^5+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4+1248*k^4*$ $9024*k^4*n^2*y^2-6848*k^4*n*y^3+1776*k^4*y^4-2016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^4*y-12016*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^3*n^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*k^5+16280*$ 41440*k^3*n^3*y^2+46160*k^3*n^2*y^3-23040*k^3*n*y^4+4160*k^3*y^5+2688*k^2*n^6-26000*k^2*n^5*y+82320*k^2*n^4*y^2-121120*k^2*n^3*y^3+89120*k^2*n^2*y^4-30880*k^2*n*y^5+3840*k^2*y^6-1728*k*n^7+19488*k*n^6*y-73920*k*n^5*y^2+ 135520*k*n^4*y^3-132160*k*n^3*y^4+67680*k*n^2*y^5-16160*k*n*y^6+1260*k*y^7+ $432*n^8-5568*n^7*y+24640*n^6*y^2-54208*n^5*y^3+66080*n^4*y^4-45120*n^3*y^5+$
$16160*n^2*y^6-2520*n*y^7+100*y^8-104*k^5*n^2+304*k^5*n*y-192*k^5*y^2+1040*k^4*n^3-1040*k^2+1040*k^2+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+1040*k^4+$ 4488*k^4*n^2*y+5488*k^4*n*y^2-2024*k^4*y^3-3640*k^3*n^4+20800*k^3*n^3*y-37600*k^3*n^2*y^2+27040*k^3*n*y^3-6600*k^3*y^4+5824*k^2*n^5-41480*k^2*n^4*y+ 99360*k^2*n^3*y^2-106000*k^2*n^2*y^3+50720*k^2*n*y^4-8600*k^2*y^5-4368*k*n^6+ $37296*k*n^5*y-111440*k*n^4*y^2+157920*k*n^3*y^3-112560*k*n^2*y^4+37540*k*n*y^5-112560*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+37540*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k*n^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2*y^4+3760*k^2$ 4370*k*y^6+1248*n^7-12432*n^6*y+44576*n^5*y^2-78960*n^4*y^3+75040*n^3*y^4-1312*k^4*y^2-3920*k^3*n^3+15480*k^3*n^2*y-17888*k^3*n*y^2+6264*k^3*y^3+ 7840*k^2*n^4-41120*k^2*n^3*y+70752*k^2*n^2*y^2-48928*k^2*n*y^3+11456*k^2*y^4 $7056*k*n^5+46200*k*n^4*y-105728*k*n^3*y^2+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n*y^4+109200*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-50736*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^3-5076*k*n^2*y^$ $8284*k*y^5+2352*n^6-18480*n^5*y+52864*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^5*y^5+2352*n^6-18480*n^5*y+52864*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^5*y+52864*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^5*y+52864*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^4*y^2-72800*n^3*y^3+50736*n^2*y^4-18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18480*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n^2*y^3+18400*n$ 16568*n*y^5+1884*y^6-12*k^5+360*k^4*n-452*k^4*y-2520*k^3*n^2+6256*k^3*n*y-
$3496*k^3*y^2+6720*k^2*n^3-24904*k^2*n^2*y+27600*k^2*n*y^2-9320*k^2*y^3-7560*k*n^4+1280*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*n^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7560*k^2*y^3-7$ $37296*k*n^3*y-61824*k*n^2*y^2+41552*k*n*y^3-9464*k*y^4+3024*n^5-18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+18648*n^4*y+$ 41216*n^3*y^2-41552*n^2*y^3+18928*n*y^4-3032*y^5+64*k^4-896*k^3*n+1064*k^3*y+ $3584*k^2*n^2-8464*k^2*n*y+4560*k^2*y^2-5376*k*n^3+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y-20416*k*n*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k*n^2*y^2+19008*k^2*y^2+19008*k^2*y^2+19008*k^2*y^2+19008*k^2*y^2+19008*k^2*y^2+19008*k^2*y^2+19008*$ $6720*k*y^3+2688*n^4-12672*n^3*y+20416*n^2*y^2-13440*n*y^3+3008*y^4-136*k^3+2008*y^4-136*k^3+2008*y^4-12672*n^3*y^4-12672*n^3+20416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416*n^2+120416$
$1088*k^2*n-1240*k^2*y-2448*k*n^2+5568*k*n*y-2920*k*y^2+1632*n^3-5568*n^2*y+1632*n^3+1632*n^2+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1632*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^3+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+1622*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+162*n^2+$ $5840*n*y^2-1888*y^3+144*k^2-648*k*n+716*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+736*y^2-76*k+152*n-16*k*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n*y+648*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2-1432*n^2$ $164*y+16)*k*(-n+y+k)^4/(2*n^2-6*n*y+5*y^2+8*n-12*y+8)/(8*k^5*y-40*k^4*n*y+16)*k*(-n+y+k)^4/(2*n^2-6*n*y+5*y^2+8*n-12*y+8)/(8*k^5*y-40*k^4*n*y+16)*k*(-n+y+k)^4/(2*n^2-6*n*y+5*y^2+8*n-12*y+8)/(8*k^5*y-40*k^4*n*y+16)*k*(-n+y+k)^4/(2*n^2-6*n*y+5*y^2+8*n-12*y+8)/(8*k^5*y-40*k^4*n*y+16)*k*(-n+y+k)^4/(2*n^2-6*n*y+5*y^2+8*n-12*y+8)/(8*k^5*y-40*k^4*n*y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*(-n+y+16)*k*$ 36*k^4*y^2+80*k^3*n^2*y-144*k^3*n*y^2+64*k^3*y^3-80*k^2*n^3*y+216*k^2*n^2*y^2-192*k^2*n*y^3+56*k^2*y^4+40*k*n^4*y-144*k*n^3*y^2+192*k*n^2*y^3-112*k*n*y^4+ $26*k*y^5-8*n^5*y+36*n^4*y^2-64*n^3*y^3+56*n^2*y^4-26*n*y^5+5*y^6-4*k^5+20*k^4*n-12*y^4-26*n^2*y^4-26*n^2*y^4-26*n^2*y^6-4*k^5+20*k^4*n-12*y^6-4*k^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6+12*y^6$ 22*k^4*y-40*k^3*n^2+88*k^3*n*y-48*k^3*y^2+40*k^2*n^3-132*k^2*n^2*y+144*k^2*n*y^2-
$52*k^2*v^3-20*k*n^4+88*k*n^3*y-144*k*n^2*y^2+104*k*n*y^3-28*k*y^4+4*n^5-22*n^4*y+104*k*n^2*y^2+104*k*n^2*y^3+104*k*n^2*y^3+104*k*n^3+y^4+104*k*n^2+y^3+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^3+y^4+104*k*n^4+104*k*n^3+y^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104*k*n^4+104$ $48*n^3*y^2-52*n^2*y^3+28*n*y^4-6*y^5+2*k^4-8*k^3*n+8*k^3*y+12*k^2*n^2-24*k^2*n*y+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*k^2+12*$ $12*k^2*y^2-8*k*n^3+24*k*n^2*y-24*k*n*y^2+8*k*y^3+2*n^4-8*n^3*y+12*n^2*y^2-8*n*y^3+2*n^4-8*n^3*y+12*n^2*y^2-8*n*y^3+2*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n^4-8*n$ $2*y^4)/(-n+k-1)^4/(n+1-y)^4$

5.1 Generalizations of Guillera's series

With the notation

$$s(x,y) = \left(x + \frac{y-1}{2}\right) \cdot {}_{5}F_{4} \left[\begin{array}{cccccc} x & x & x & x & x + \frac{y+1}{2} & 1 \\ x+y & x+y & x+y & x+y & x + \frac{y-1}{2} \end{array}; -1 \right],$$

Theorem 6 gives us that

$$s(x,y) = x + \frac{y-1}{2} + \frac{x^4}{(x+y)^4} s(x+1,y),$$

$$s(x,y) = \frac{5y^2 + 6(x-1)y + 2(x-1)^2}{4(2y-1)} - \frac{y^5}{2(2y-1)(x+y)^4} s(x,y+1).$$

Note that for $x = \frac{1}{2}, y = \frac{3}{2}, s(x, y)$ can be calculated from its partial sums:

$$s(\frac{1}{2}, \frac{3}{2}) = \frac{1}{4} \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{8n} {2n \choose n}^4 \frac{4n+3}{(n+1)^4} = 4 - \frac{32}{\pi^2}$$

If we try to accelerate this series with the first of the recurrences, we get the same series. With the second one, we get Guillera's formula shown in (3) [14].

If we use both the recurrences for x = 1, y = 1, we get the following result:

$$\zeta(3) = \frac{1}{2} \sum_{n=1}^{\infty} (-1)^{n-1} \frac{205n^2 - 160n + 32}{n^5 \binom{2n}{n}^5}$$

a series which can be found in Lupaş's paper [25]. Now, let us further simplify our expression for the latter recursion given in Theorem 6, letting h, r_1 , and r_2 be as below:

$$h(x,y,k) = \frac{\left(x + \frac{y-1}{2}\right) ((x)_k)^4 \left(x + \frac{y+1}{2}\right)_k}{\left(x + \frac{y-1}{2}\right)_k ((x+y)_k)^4},$$

$$r_1(x,y) = \frac{6(x-1)y + 2(x-1)^2 + 5y^2}{4(2y-1)},$$

$$r_2(x,y) = -\frac{y^5}{2(2y-1)(x+y)^4}.$$

So, we obtain that

$$\sum_{k=0}^{\infty} h(x,y,k) = r_1(x,y) + r_2(x,y) \sum_{k=0}^{\infty} h(x,y+1,k),$$
(22)

leading us to the following result.

Theorem 7. The infinite series

$$\sum_{k=0}^{\infty} h(x, y, k)$$

may be accelerated as

$$\sum_{j=0}^{m-1} \left(\prod_{i=0}^{j-1} r_2(x, y+i) \right) r_1(x, y+j) + \left(\prod_{i=0}^{m-1} r_2(x, y+i) \right) \sum_{k=0}^{\infty} h(x, y+m, k)$$

for natural numbers m, with

$$\left(\prod_{i=0}^{m-1} r_2(x, y+i)\right) \sum_{k=0}^{\infty} h(x, y+m, k) \to 0$$
 (23)

as $m \to \infty$, under the assumption that the above series converge, for given values of x and y.

Proof. The above acceleration formula follows from repeated applications of (22), and the limiting property in (23) follows in a direct way from our definitions for r_2 and h, under the given assumption.

The above Theorem may be regarded as generalization of (3), in view of our proof of (3), along with the following results.

Example 4. Setting $x = \frac{3}{2}$ and $y = \frac{3}{2}$, our series acceleration formula given in Theorem 7 yields the following:

$$\frac{128}{\pi^2} = \sum_{j=0}^{\infty} (-1)^j \left(\frac{1}{2}\right)^{12j} {2j \choose j}^5 \frac{20j^2 + 32j + 13}{(j+1)^4}.$$

Example 5. Setting $x = \frac{5}{2}$ and $y = \frac{3}{2}$, we obtain:

$$\frac{2048}{81\pi^2} = \sum_{j=0}^{\infty} (-1)^j \left(\frac{1}{2}\right)^{12j} {2j \choose j}^5 \frac{20j^2 + 56j + 41}{(j+1)^4 (j+2)^4}.$$

16

Example 6. Setting $x = \frac{1}{2}$ and y = 1, this gives us that

$$28\zeta(3) = \sum_{j=1}^{\infty} \frac{(-1)^j 2^{8j} \left(-10j^2 + 6j - 1\right)}{j^5 \binom{2j}{j}^5}.$$

Example 7. If we use both recurrences from Theorem 6 one after the other, with $x = \frac{3}{2}$ and $y = \frac{3}{2}$, we obtain Guillera's series from [15]:

$$\frac{128}{\pi^2} = \sum_{j=0}^{\infty} (-1)^j \left(\frac{1}{2}\right)^{20j} {2j \choose j}^5 (820j^2 + 180j + 13).$$

6 Non-computer proofs

As an example, we prove the following theorem, which can be found in [23]. The proof given here is much shorter than the one in [23].

Theorem 8. The recurrence

$$_{3}F_{2}\begin{bmatrix} x & x & 1 \\ x + y & x + y \end{bmatrix}$$
; $_{1}$ = $\frac{2x + 3y - 2}{2(2y - 1)} + \frac{y^{3}}{2(2y - 1)(x + y)^{2}} \cdot {}_{3}F_{2}\begin{bmatrix} x + 1 & x + 1 & 1 \\ x + 1 + y & x + 1 + y \end{bmatrix}$; $_{1}$

hold true, if the above series converge.

Or, with the notation

$$s(x,y) = {}_{3}F_{2} \begin{bmatrix} x, x, 1 \\ x+y, x+y \end{bmatrix} = \sum_{n=0}^{\infty} \frac{(x)_{n}^{2}}{(x+y)_{n}^{2}} :$$

$$s(x,y) = \frac{2x+3y-2}{2(2y-1)} + \frac{y^{3}}{2(2y-1)(x+y)^{2}} \cdot s(x,y+1)$$

Proof. The proof is based on the following identity, which is easy to prove:

$$\frac{(x+n)^2}{(x+y+n)^2} = \frac{2n+2x-y}{2(2y-1)} - \frac{2(n+1)+2x-y}{2(2y-1)} \frac{(x+n)^2}{(x+y+n)^2} + \frac{y^3}{2(2y-1)(x+y+n)^2}.$$

We will use the notation $f(x,y,n) = \frac{2n+2x-y}{2(2y-1)}$. We rewrite the series for s(x,y) using this identity:

$$\begin{split} s(x,y) &= 1 + \sum_{n=0}^{\infty} \frac{(x)_{n+1}^2}{(x+y)_{n+1}^2} = 1 + \sum_{n=0}^{\infty} \frac{(x)_n^2}{(x+y)_n^2} \cdot \frac{(x+n)^2}{(x+y+n)^2} \\ (\star) &= 1 + \sum_{n=0}^{\infty} \frac{((x)_n)^2}{((x+y)_n)^2} \left(f(x,y,n) - f(x,y,n+1) \frac{(x+n)^2}{(x+y+n)^2} + \frac{y^3}{2(2y-1)(x+y+n)^2} \right) \\ &= 1 + f(x,y,0) + \frac{y^3}{2(2y-1)} \sum_{n=0}^{\infty} \frac{(x)_n^2}{(x+y)_{n+1}^2} \\ &= \frac{2x+3y-2}{2(2y-1)} + \frac{y^3}{2(2y-1)(x+y)^2} \sum_{n=0}^{\infty} \frac{((x)_n)^2}{((x+y+1)_n)^2} \end{split}$$

Note that (\star) simplifies as a consequence of telescoping, assuming that $\lim_{n\to\infty} \frac{((x)_n)^2}{((x+y)_n)^2} f(x,y,n) = 0$.

Remark 1. While writing this paper, we noted that for s(x, y) above:

$$s(-\frac{1}{2}, \frac{3}{2}) = \sum_{n=0}^{\infty} \frac{\binom{2n}{n}^2}{(2n-1)^2 2^{4n}} = \frac{4}{\pi}$$

(Forsyth's series, see [23]). Using Theorem 8 to accelerate this series, we get Gosper's well known series from 1974 [13]:

$$12\sum_{n=1}^{\infty} \left(\frac{1}{2}\right)^{8n} {2n \choose n}^3 n^2 (2n-1) = \frac{4}{\pi}.$$

Note that this series is equivalent with Ramanujan's well known series:

$$\sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^{8n} {2n \choose n}^3 (6n+1) = \frac{4}{\pi}$$

since the difference of both series can be written as

$$\lim_{k \to +\infty} \sum_{n=0}^{k-1} \left(\frac{1}{2}\right)^{8n} {2n \choose n}^3 (24n^3 - 12n^2 - 6n - 1) = -\lim_{k \to +\infty} \frac{32k^3 {2k \choose k}^3}{2^{8k}} = 0.$$

6.1 Proof of Theorem 1

We use the notation from subsection 2.1. The proof is similar to the previous one, but now based on the identity:

$$\frac{(x+n)(x+n+1)}{(x+y+n)(x+y+n+1)} = \frac{2n+2x-y+1}{2(2y-1)} - \frac{2n+2+2x-y+1}{2(2y-1)} \frac{(x+n)(x+n+1)}{(x+y+n)(x+y+n+1)} + \frac{(y-1)y(y+1)}{2(2y-1)} \frac{1}{(x+y+n)(x+y+n+1)}.$$

6.2 Proof of Theorem 2

This is can be done using a reindexing argument.

6.3 Proof of Theorem 4

For the second part we use:

$$s(x,y) = \sum_{n=0}^{\infty} (-1)^n \frac{(x)_n^2}{(y)_n^2}$$

and the identity:

$$(x+n)^2 = h(x,y,n) + h(x,y,n+1)\frac{(x+n)^2}{(x+y+n+1)^2} + \frac{y(1+y)^3}{4(x+y+n+1)^2}$$

with
$$h(x, y, n) = \frac{2n^2 + 2(2x+y)n + 2x^2 + 2xy - y^2 - y}{4}$$
.

6.4 Proof of Theorem 5

Note that in this case:

$$s(x,y) = \sum_{n=0}^{\infty} \frac{(x)_n^3 (x + \frac{y+1}{2})_n}{(x+y)_n^3 (x + \frac{y+1}{2})_{n-1}}$$

For the proof of the first part only reindexing is needed. For the second part, we rewrite as before, and use the identity:

$$\frac{(x+n)^3(x+\frac{y+1}{2}+n)}{(x+y+n)^3} = \frac{n+x-y}{2} + \frac{n+1+x-y}{2} \frac{(x+n)^3}{(x+y+n)^3} + y^3 \cdot \frac{(x+\frac{y+2}{2}+n-1)}{(x+y+n)^3}.$$

6.5 Proof of Theorem 6

For the second part we use:

$$s(x,y) = \sum_{n=0}^{\infty} \frac{(x)_n^4 (x + \frac{y+1}{2})_n}{(x+y)_n^4 (x + \frac{y+1}{2})_{n-1}}$$

and the identity:

$$\frac{(x+n)^4(x+\frac{y+1}{2}+n)}{(x+y+n)^4} = g(x,y,n) - g(x,y,n+1)\frac{(x+n)^4}{(x+y+n)^4} - \frac{y^5}{2(2y-1)} \cdot \frac{(x+\frac{y+2}{2}+n-1)}{(x+y+n)^4}$$

with
$$g(x, y, n) = \frac{2n^2 + 2(2x - y)n + 2x^2 - 2xy + y^2}{4(2y - 1)}$$

References

- [1] J. A. Adell and A. Lekuona, Acceleration methods for series: a probabilistic perspective, *Mediterr. J. Math.* **13** (2016), 5063–5076.
- [2] J. A. ADELL AND A. LEKUONA, Dirichlet's eta and beta functions: concavity and fast computation of their derivatives, *J. Number Theory* **157** (2015), 215–222.
- [3] W. N. Bailey, "Generalized Hypergeometric Series". Cambridge Univ. Press, 1935.
- [4] J. M. Campbell, A Wilf–Zeilberger–based solution to the Basel problem with applications, *Discrete Math. Lett.* **10** (2022), 21–27.
- [5] J. M. Campbell, A WZ proof for a Ramanujan-like series involving cubed binomial coefficients, J. Difference Equ. Appl. 27 (2021), 1507–1511.
- [6] J. M. Campbell, WZ Proofs of Identities from Chu and Kılıç, with Applications, to appear in *Appl. Math. E-Notes*.
- [7] E. CATALAN, Mémoire sur la transformation des séries et sur quelques intégrales définies, Mémoires couronnés et mémoires des savants étrangers 33 (1867), 1–50.
- [8] W. Chu, Abel's method on summation by parts and hypergeometric series, J. Difference Equ. Appl. 12 (2006), 783–798.

- [9] W. Chu and E. Kiliç, Binomial sums involving Catalan numbers, *Rocky Mountain J. Math.* **51** (2021), 1221–1225.
- [10] W. Chu and W. Zhang, Accelerating Dougall's $_5F_4$ -sum and infinite series involving π , Math. Comp. 83 (2014), 475–512.
- [11] S. B. EKHAD AND D. ZEILBERGER, A WZ proof of Ramanujan's formula for π , in "Geometry, analysis and mechanics". World Sci. Publ., River Edge, NJ, 1994.
- [12] E. Fabry, "Théorie des séries à termes constants: applications aux calculs numériques". Hachette Livre Bnf, 2020.
- [13] R. W. Gosper, Acceleration of series, Tech. Report AIM-304, MIT, Cambridge, MA, USA, 1974.
- [14] J. Guillera, About a new kind of Ramanujan-type series, Experiment. Math. 12 (2003), pp. 507—510.
- [15] J. Guillera, Hypergeometric identities for 10 extended Ramanujan-type series, Ramanujan J., 15(2008), 219–234.
- [16] KH. HESSAMI PILEHROOD AND T. HESSAMI PILEHROOD, Bivariate identities for values of the Hurwitz zeta function and supercongruences, *Electron. J. Combin.* **18** (2011), Paper 35, 30.
- [17] KH. HESSAMI PILEHROOD AND T. HESSAMI PILEHROOD, Generating function identities for $\zeta(2n+2)$, $\zeta(2n+3)$ via the WZ method, *Electron. J. Combin.* **15** (2008), Research Paper 35, 9.
- [18] Kh. Hessami Pilehrood and T. Hessami Pilehrood, Series acceleration formulas for beta values, *Discrete Math. Theor. Comput. Sci.* **12** (2010), 223–236.
- [19] KH. HESSAMI PILEHROOD, T. HESSAMI PILEHROOD, AND R. TAURASO, Congruences concerning Jacobi polynomials and Apéry-like formulae, *Int. J. Number Theory* 8 (2012), 1789–1811.
- [20] K. Knopp, "Theory and Application of Infinite Series". Dover, New York, 1990.
- [21] KH. HESSAMI PILEHROOD, T. HESSAMI PILEHROOD, AND R. TAURASO, New properties of multiple harmonic sums modulo p and p-analogues of Leshchiner's series, Trans. Amer. Math. Soc. 366 (2014), 3131–3159.
- [22] E. E. Kummer, Eine neue Methode, die numerischen Summen langsam convergirender Reihen zu berechnen, *J. reine angew. Math.* **16** (1837), pp. 206–214.
- [23] P. Levrie and A. S. Nimbran, From Wallis and Forsyth to Ramanujan, *Ramanujan J.* 47 (2018), 533–545.
- [24] F. M. S. Lima, A rapidly converging Ramanujan-type series for Catalan's constant, (2012), arXiv:1207:3139.
- [25] A. Lupaş, Formulae for some classical constants, in "RoGer 2000—Braşov". Gerhard-Mercator-Univ., Duisburg, 2000.

- [26] O. Marichev, J. Sondow, and E. W. Weisstein, Catalan's Constant, From MathWorld-A Wolfram Web resource. Available online https://mathworld.wolfram.com/CatalansConstant.html
- [27] M. OJANGUREN, From Wallis to Ramanujan, in the company of R. W. Gosper, *Expo. Math.* **36** (2018), 325–342.
- [28] M. Petkovšek, H. S. Wilf, and D. Zeilberger, "A=B". A K Peters, Ltd., Wellesley, MA, 1996.
- [29] X. Wang and W. Chu, Series with harmonic-like numbers and squared binomial coefficients, Rocky Mountain J. Math..
- [30] W. Wang and C. Xu, Alternating multiple zeta values, and explicit formulas of some Euler-Apéry-type series, *European J. Combin.* **93** (2021), Paper No. 103283, 25.

Paul Levrie
Faculty of Applied Engineering
UAntwerpen, Groenenborgerlaan 171, 2020 Antwerp, Belgium
and
Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee (Leuven)
Belgium, P. O. Box 2402
paul.levrie@kuleuven.be

John M. Campbell
Department of Mathematics and Statistics
York University
4700 Keele St, Toronto, Ontario, Canada M3J 1P3
jmaxwellcampbell@gmail.com