Задача D1. Календар

Училището в село Каръшко, което както е известно, признателното ръководство кръсти на наше име: "Програмистите на България", на 20.11.2005 г. ще празнува своя 70-ти юбилей. По случай празника директорът решил да организира състезание, като всеки един от участниците получил следната задача: по зададена дата (ден, месец, година) и какъв ден от седмицата е била тя, да се отпечата календарът за определен месец от същата година.

Умко (еднин от най-умните и мързеливи участници) ви моли да му помогнете да спечели състезанието, като напишете програма **CALENDAR**, която прочита от клавиатурата данните, зададени от директора и отпечатва календара на екрана на компютъра.

Програмата приема от първия ред на стандартния вход четири цели числа: деня, месеца, годината и деня от седмицата за известната дата (понеделник се отбелязва с 1, вторник – с 2, сряда – с 3 и т.н.), а от втория ред – само едно число – месеца, чийто календар трябва да се отпечата.

На стандартния изход, програмата отпечатва календара по следния начин: на първия ред се отпечатват първите букви на дните (на латиница) на седмицата, отделени с по два интервала. На следващите редове се отпечатват дните на зададения месец, започващи от 1, като всяка дата е под съответния ден от седмицата за този месец. Едноцифрените дни са разделени помежду си с по два интервала, а двуцифрените – с по един.

Примерен вход:

```
19 11 2005 6
1
```

Примерен изход:

```
P   V   S   C   P   S   N
                    1   2
3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31
```

Решение:

За да отпечатаме календара на даден месец трябва да определим какъв ден от седмицата е първият му ден. За целта трябва да се пресметне броят на дните между известната дата и първия ден на посочения месец. Тук трябва да обърнем внимание, че ако месецът е преди датата броя на дните се смята по един начин, а ако е след датата – по друг. Използва се масив, съдържащ броя на дните на всеки един от месеците, което улеснява пресмятането. Естествено, че трябва да съобразим да променим броя на дните на месец февруари за високосните години.

След като пресметнем броя на дните, чрез деление на 7 с остатък пресмятаме в кой ден от седмицата е първият ден от търсения месец. Когато получим това, остава само с помощта на един цикъл да отпечатаме и самия календар.

Ето една примерна програма, която решава горната задача:

```
#include <iostream.h>
void main()
{
  int d,m,g,d1,m1,sum=0,os,pom,i,j,br=0,
  a[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
   cin>>d>>m>>g>>d1>>m1;
   if(g%4==0)a[2]=29;
   if(m>=m1)
   {
```

```
     for(i=m-1;i>=m1;i--)sum+=a[i];
     sum+=d;
     os=sum%7;
      if(d1>os)pom=(d1-os)+1;
      else pom=(7-(os-d1)+1);
      pom=(pom%7)?pom%7:7;
     }
     else
     {
      for(i=m+1;i<m1;i++)sum+=a[i];
      sum+=(a[m]-d)+1;
      os=sum%7;
      pom=(os+d1)%7;
      if(pom==0)pom=7;
     }
   cout<<"P  V  S  C  P  S  N\n";
   for(i=1;i<=pom-1;i++)cout<<"   ";
   for(i=1;i<=(7-(pom-1));i++)cout<<i<<"  ";
   cout<<endl;
   for(j=i;j<=a[m1];j++)
    if(br!=6)
     {
       if(j>=10){cout<<j<<" ";br++;}
       else {cout<<j<<"  ";br++;}
     }
    else
     if(j>=10){cout<<j<<"\n";br=0;}
     else {cout<<j<<"  "<<"\n";br=0;}
    cout<<endl;
}
```

Задача D2. Милионер

Впечатлен от историята за забогатяване на Хенри Форд (започнал от една ябълка, продал я, купил 2 ябълки и т.н.), която госпожата в училището "Програмистите на България" разказала, Умко, който освен добър ученик е и футболен фен, си изградил следната стратегия за забогатяване, залагайки в „ Еврофутбол":

- За всеки нов тираж се прави един залог с коефициенти 2 или 3. Това значи, че ако заложим К лв. **печалбата** ще бъде съответно 2.К лв или 3.К лв.
- За всеки тираж заложената сума се определя от **печалбата** от предходен тираж увеличена с 1 лв. Всяка **печалба** се залага два пъти: веднъж с коефициент 2 и веднъж с коефициент 3. Спазва се правилото, че новата **печалба** не трябва да е по-малка от предходната.

Помогнете на Умко да намери след колко залагания, следвайки горната стратегия, ще стане милионер, ако в началото заложи 1 лв. Напишете програма **milioner.exe**, която въвежда цяло число ($1<n\leq 440000$) и извежда след най-малко колко залагания ще се получи **печалба** не по-малка от **n** и каква ще е стойността на тази **печалба**.

Вход: `9`
Изход: `5 9`
Печалбите са съответно: `2 3 6 8` **`9`**
Вход:25
Изход:11 26
Печалбите са съответно: `2 3 6 8 9 12 14 18 20 21` **`26`**

Коментар: Задачата се състои в генериране на елементи на множество по даден алгоритъм. Предложеното решение извършва тази генерация итерационно, като се използуват вече генерираните елементи съхранявани в масив и се избира по-малкия от новополучените. Използуват се два брояча за генерационните клонове за запазване на възходящия ред на получаване на елементите. С цел икономия на памет елементите неучастващи в следващите стъпки се изтриват. Стандартното решение ще покрие само част от тестовете.

```
#include <iostream.h>
void main()
{long y=0,i=1,i2=1,i3=1,k,n,x2,x3,
    x[15580] ;

 cin>>n;x[1]=1;
 do
 {   y++;x2=2*x[i2]+1;x3=3*x[i3]+1;
    i++;
    if (x2<=x3) { x[i]=x2;i2++; }
    else { x[i]=x3;
         for (k=2;k<=i;k++) x[k-1]=x[k];
         i--;i2--;
       }
  }
 while(x[i]-1<n);
cout<<y<<" "<<x[i]-1;
}
```

Задача D3. Думи в текст

За домашно по информатика учителката дала на Умко интересна задача, но той както обикновенно “няма време” да я реши и пак ще прибегне до вашата помощ. Все пак училището му носи вашето име: “Програмистите на България”. В задачата се иска да се напише програма **WORDS**, която прочита от първия ред на стандартния вход дума, съдържаща не повече от 20 знака, а от втория – произволен текст, съдържащ не повече от 1000 знака. Програмата трябва да отпечата номера на началния и крайния знак на най-дългата последователност от знаци в текста, които принадлежат на думата. Програмата трябва да може да различава малка от главна буква. Ако в текста не се среща нито една буква от думата, програмата извежда 0.

Примерен вход**:**
```
vaza
Programistite na Bulgaria se sastezavat.
```
Примерен изход**:**
```
35 38
```

Примерен вход**:**
```
ana
Atanas yade ananas
```
Примерен изход**:**
```
13 17
```

Примерен вход**:**
```
a
aaaa
```
Примерен изход:
```
1 4
```

Решение:

1. Необходими величини:

Две стрингови променливи, съответно за думата и текста.

```
char d[20], s[1000];
```

Пет целочислени променливи, съответно за дължината на думата и текста, началото и дължината на най-дългата последователност от знаци на думата, които се съдържат в текста, както и за началото и дължината на текущата последователност. Началната стойност на всяка една от тези променливи трябва да бъде 0.

```
int n, mn=0, md=0, tn=0, td=0, i, j;
```

Допълнително се декларират няколко работни променливи, които ще се използват за управление на циклите.

За определяне на принадлежността на даден знак към думата се използва помощен масив с 256 знака. Всеки елемент на този масив има стойност 0 ако знака със съответния код не е в думата и 1 – в противен случай. Този масив също се нулира в началото на програмата.

```
int b[256];
for(i=0; i<256; i++) b[i] = 0;
```

2. Въвежда се думата, намира се нейната дължина и се запълва масива b.

```
cin >> d;
n = strlen(d);
for(i=0; i<n; i++) b[d[i]] = 1;
```

3. Въвежда се текста и се определя дължината му:

```
cin.getline(s, 1000, '\n');
n = strlen(s);
```

4. Обхожда се текста и се намира най-дългата паследователност от от знаци на думата, които се съдържат в текста, като за всеки един от знаците на s, се проверява дали принадлежи на думата, т.е. дали съответното b е 1, при което се увеличава броя на буквите в текущата последователност от знаци, принадлежащи на думата. Ако знакът не принадлежи на думата са възможни два варианта: или това е края на текущата последователност и тогава трябва да проверим дали тя не е по-дълга от намерената до момента; или това е просто поредния знак от текста, който не принадлежи на думата и няма нужда от обработка.

```
for(i=0;i<n;i++)
     if(b[s[i]]){ if(!td)tn = i; td++;}
        else
          if(td)
            {
              if(td > md){md = td; mn = tn;};
              td=0;
            }
```

5. За да не се пропусне случая когато най-дългата последователност остава последна в текста, обработката за най-дълга последователност се повтаря след като цикъла завърши:

```
if(td > md){md = td; mn = tn;};
```

6. Накрая се извежда началото и края на намерената най-дълга последователност.

```
cout << mn + 1 << ' ' << mn + md << endl;
```