



Test grilă...

BAC INFO' 2003

Chiar dacă sunt disponibile puține informații referitoare la organizarea examenului de Bacalaureat la disciplina informatică, vă prezentăm o variantă orientativă, neoficială, a unui test grilă pentru acest examen. Testul este disponibil online la adresa <http://www.ginfo.ro/diverse/bac.shtml>.

Întrebarea #1:

Care dintre următoarele afirmații este falsă?

- a) Un algoritm are, prin definiție, un număr finit de pași.
- b) Poate fi găsit un algoritm pentru rezolvarea oricărei probleme.
- c) Operațiile corespunzătoare unui algoritm trebuie executate într-o anumită ordine.
- d) Pentru o anumită problemă ar putea exista mai mulți algoritmi de rezolvare.

Întrebarea #2:

Care dintre următoarele afirmații este adevărată?

- a) O expresie trebuie să conțină întotdeauna cel puțin o variabilă
- b) Valoarea unei variabile se poate modifica pe parcursul executării unui algoritm
- c) O expresie nu poate conține decât cel mult un operator
- d) O operație constă întotdeauna în efectuarea unor calcule folosind mai multe variabile

Întrebarea #3:

Corpul unei structuri repetitive anterior condiționate se execută:

- a) exact o dată
- b) cel mult o dată
- c) cel puțin o dată
- d) de zero, una sau mai multe ori

Întrebarea #4:

Versiune Pascal:

Care dintre următoarele cuvinte nu este rezervat în limbajul Pascal:

- a) for
- b) while
- c) write
- d) if

Versiune C:

Care dintre următoarele cuvinte nu este rezervat în limbajul C:

- a) for
- b) while
- c) printf
- d) if

Întrebarea #5:

Versiune Pascal:

Care dintre următoarele cuvinte cheie este folosit pentru declararea unui tip în limbajul Pascal?

- a) type
- b) record
- c) set
- d) typedef

Versiune C:

Care dintre următoarele cuvinte cheie este folosit pentru declararea unui tip în limbajul C?

- a) typedef
- b) record
- c) struct
- d) type

Întrebarea #6:

Versiune Pascal:

Care este modalitatea prin care se declară în limbajul Pascal o variabilă a de tip caracter și cum i se atribuie acesteia ca valoare litera H?

- | | |
|-----------------------------|--------------------|
| a) Declarație: a:character; | Atribuire: a:='H'; |
| b) Declarație: a:char; | Atribuire: a:='H'; |
| c) Declarație: a:char; | Atribuire: a:=H; |
| d) Declarație: a:character; | Atribuire: a:=H; |

Versiune C:

Care este modalitatea prin care se declară în limbajul C o variabilă a de tip caracter și cum i se atribuie acesteia ca valoare litera H?

- | | |
|-----------------------------|-------------------|
| a) Declarație: character a; | Atribuire: a='H'; |
| b) Declarație: char a; | Atribuire: a='H'; |
| c) Declarație: char a; | Atribuire: a=H; |
| d) Declarație: character a; | Atribuire: a=H; |

Întrebarea #7:

Este adevărat că o funcție definită de utilizator trebuie să aibă întotdeauna cel puțin un parametru?

- a) Da
- b) Nu
- c) Da, dar numai dacă nu returnează nici o valoare
- d) Da, dar numai dacă returnează cel puțin o valoare

Întrebarea #8:

Versiune Pascal:

Care dintre următoarele instrucțiuni poate fi folosită pentru a atribui unei variabile a codul ASCII al caracterului *?

- a) `a := ord('*');` b) `a := '*';`
- c) `a := chr('*');` d) `a := ascii(*);`

Versiune C:

Care dintre următoarele instrucțiuni poate fi folosită pentru a atribui unei variabile a codul ASCII al caracterului *?

- a) `a = '*';` b) `a = ord('*');`
- c) `a = chr('*');` d) `a = ascii(*);`

Întrebarea #9:

Versiune Pascal:

Care dintre următoarele expresii poate fi utilizată pentru a verifica dacă valoarea variabilei a este impară?

- a) `a or 1 = 1` b) `a xor 1 = 1`
- c) `a and 1 = 1` d) `a not 1 = 1`

Versiune C:

Care dintre următoarele expresii poate fi utilizată pentru a verifica dacă valoarea variabilei a este impară?

- a) `a | 1` b) `a ^ 1`
- c) `a & 1` d) `a ~ 1`

Întrebarea #10:

Versiune Pascal:

Care dintre următoarele afirmații este falsă în cazul limbajului Pascal?

- a) Elementele unui tablou au același tip.
- b) Tabloul este o structură statică de date.
- c) În declarația unui tablou trebuie folosit cuvântul cheie `array`.
- d) Primul element al unui tablou are întotdeauna indicele 1.

Versiune C:

Care dintre următoarele afirmații este falsă în cazul limbajului C?

- a) Elementele unui tablou au același tip.
- b) Tabloul este o structură statică de date.
- c) În declarația unui tablou nu trebuie folosit cuvântul cheie `array`.
- d) Primul element al unui tablou are întotdeauna indicele 1.

Întrebarea #11:

Versiune Pascal:

Care dintre următoarele funcții poate fi utilizată pentru determinarea numărului de caractere dintr-un string?

- a) `strlen` b) `length` c) `len` d) `size`

Versiune C:

Care dintre următoarele funcții poate fi utilizată pentru determinarea numărului de caractere dintr-un string?

- a) `size` b) `length`
- c) `strlen` d) `len`

Întrebarea #12:

Versiune Pascal:

Care dintre declarații reprezintă un articol care conține informațiile referitoare la un punct în plan ale cărui coordonate sunt numere reale?

- a) `var a:record x,y:real end;`
- b) `var a:record begin x,y:real end;`
- c) `var x,y:real;`
- d) `var a:record a.x,a.y:real end;`

Versiune C:

Care dintre declarații reprezintă un articol care conține informațiile referitoare la un punct în plan ale cărui coordonate sunt numere reale?

- a) `struct {float x,y;}a;`
- b) `struct {float x} {float y}a;`
- c) `float x, y;`
- d) `struct {float a.x,a.y;}a;`

Întrebarea #13:

Care dintre următoarele afirmații este adevărată?

- a) Un fișier text poate conține doar litere ale alfabetului latin.
- b) Înainte de a prelua date dintr-un fișier text, acesta trebuie deschis pentru citire.
- c) Nu există posibilitatea de a adăuga litere la sfârșitul unui fișier text.
- d) Un fișier text nu poate conține mai mult de 65535 de caractere.

Întrebarea #14:

Versiune Pascal:

Care este cuvântul cheie cu ajutorul căruia se specifică faptul că o anumită variabilă reprezintă un fișier text?

- a) `text`
- b) `file`
- c) `textfile`
- d) nu există nici un astfel de cuvânt cheie

Versiune C:

Care este cuvântul cheie cu ajutorul căruia se specifică faptul că o anumită variabilă reprezintă un fișier text?

- a) `TEXT`
- b) `FILE*`
- c) `TEXT_FILE`
- d) nu există nici un astfel de cuvânt cheie

Întrebarea #15:

Versiune Pascal:

Prin intermediul cărei proceduri se realizează corespondența dintre o variabilă și un anumit fișier stocat pe disc?

- a) `assign` b) `reset`
- c) `open` d) `fileopen`

Versiune C:

Prin intermediul cărei funcții se realizează corespondența dintre o variabilă și un anumit fișier stocat pe disc?



buc



- a) fopen
- b) fclose
- c) assign
- d) fopen

Întrebarea #16:

Ce realizează următorul algoritm descris în pseudocod?

```
citește m
s = 0
cât timp m > 0 execută
    s = s * 10 + rest[m/10]
    m = [m / 10]
sfârșit cât timp
```

- scrie s
- a) determină suma cifrelor unui număr
 - b) verifică dacă un număr se divide cu 10
 - c) determină inversul unui număr (numărul obținut prin citirea cifrelor de la dreapta la stânga)
 - d) s va avea aceeași valoare cu m

Întrebarea #17:

Care dintre numere va fi recunoscut ca fiind prim (deși, de fapt nu este) dacă se utilizează următorul algoritm de verificare a primalității?

```
citește nr
prim = adevărat
dacă nr < 2 atunci
    prim = fals
sfârșit dacă
dacă rest[nr/2] = 0 atunci
    dacă nr > 2 atunci
        prim = fals
    sfârșit dacă
sfârșit dacă
d = 3;
rad = nr^(1/2)          /** rad este radical din nr **/
cât timp (d < rad) și prim execută
    dacă rest[nr/d] = 0 atunci
        prim = fals
    altfel
        d = d + 2
    sfârșit dacă
sfârșit cât timp
dacă prim atunci
    scrie 'Numărul este prim!'
altfel
    scrie 'Numărul nu este prim!'
sfârșit dacă
```

- a) algoritmul va intra în ciclu infinit pentru orice număr neprim
- b) 17
- c) 6
- d) 25

Întrebarea #18:

Care este cel mai mic termen al șirului *Fibonacci*, format din cel puțin trei cifre, și care este prim?

- a) 233
- b) 144
- c) 377
- d) șirul *Fibonacci* nu conține numere prime.

Întrebarea #19:

Care este elementul neutru al operației de determinare a minimului a două numere?

- a) 0
- b) ∞
- c) $-\infty$
- d) această operație nu are element neutru

Întrebarea #20:

În cazul sortării prin inserție a unor numere întregi la fiecare pas se determină:

- a) poziția elementului curent în șirul parțial sortat la pași anteriori
- b) elementul minim al șirului nesortat
- c) elementul maxim al șirului nesortat
- d) nici una dintre cele trei variante anterioare

Întrebarea #21:

Care dintre următoarele condiții trebuie îndeplinite pentru ca două șiruri să poată fi interclasate?

- a) șirurile trebuie să aibă același număr de elemente
- b) șirurile trebuie să fie ordonate în funcție de un același criteriu
- c) numărul total de elemente din cele două șiruri trebuie să fie par
- d) nici una dintre cele trei variante anterioare

Întrebarea #22:

Dacă se dorește căutarea unui număr într-un șir ordonat de 1000 de elemente, folosind algoritmul căutării binare, atunci numărul maxim al pașilor efectuați va fi:

- a) cuprins între 1 și 6
- b) cuprins între 7 și 13
- c) cuprins între 14 și 20
- d) mai mare decât 20.

Întrebarea #23:

Parametrii unui subprogram pot fi transmiși prin:

- a) valoare formală sau valoare efectivă
- b) referință formală sau referință efectivă
- c) valoare sau referință
- d) numai prin referință formală

Întrebarea #24:

Care dintre următoarele afirmații este adevărată?

- a) un subprogram poate avea cel mult 12 parametri
- b) un subprogram trebuie să conțină cel puțin o instrucțiune repetitivă
- c) un subprogram nu poate fi apelat din cadrul unui alt subprogram
- d) nici una dintre cele trei

Întrebarea #25:

Versiune Pascal:

Care dintre declarații este validă pentru a declara o procedură în limbajul Pascal?

- a) procedure proc(var a:byte);
- b) procedure proc(a:array[1..100] of integer);
- c) procedure proc(a:byte):byte;
- d) procedure procedure(a:byte);

Versiune C:

Care dintre declarații este validă pentru a declara o procedură în limbajul C?

- a) în limbajul C nu există proceduri
 b) `int proc(char a)`
 c) `void proc(void)`
 d) `int proc(a:char):char`

Întrebarea #26:

Versiune Pascal:

Ce se va afișa în urma execuției programului:

```
program Valori;
var a,b:integer;
```

```
procedure aduna;
var a:integer;
begin
  a:=a+2;
  b:=b+2;
end;
```

```
begin
  a:=3;
  b:=3;
  aduna;
  writeln(a,' ',b);
end.
```

- a) 5 5 b) 5 3 c) 3 5 d) 3 3

Versiune C:

Ce se va afișa în urma execuției programului:

```
#include <stdio.h>
int a,b;

void aduna(void) {
  int a;
  a+=2;
  b+=2;
}

void main(void) {
  a = b = 3;
  aduna();
  printf("%d %d\n",a,b);
}
```

- a) 5 5 b) 5 3 c) 3 5 d) 3 3

Întrebarea #27:

Recursivitatea poate fi:

- a) directă sau indirectă b) interactivă sau pasivă
 c) completă sau incompletă d) formală sau efectivă

Întrebarea #28:

Versiune Pascal:

Ce valoare returnează următoarea funcție:

```
function f(n:integer);
begin
  if n>0 then f:=n*f(n-1);
  else f:=0;
end.
```

- a) $n!$ b) $(n-1)!$
 c) 0 d) se va intra în ciclu infinit

Versiune C:

Ce valoare returnează următoarea funcție:

```
int f(int n){
  return (n)?n*f(n-1):0;
}
```

a) $n!$ b) $(n-1)!$
 c) 0 d) se va intra în ciclu infinit

Întrebarea #29:

În care dintre următoarele situații este recomandată folosirea metodei *backtracking*:

- a) când problema este definită recursiv
 b) când trebuie determinate toate soluțiile unei probleme
 c) când trebuie găsit cel mai rapid algoritm
 d) nici una dintre cele trei.

Întrebarea #30:

Versiune Pascal:

Ce realizează următorul program:

```
program Backtracking;
var a:array[0..100] of integer;
    m,n:integer;
```

```
procedure ScribeSolutie;
var i:integer;
begin
  write(a[1]);
  for i:=2 to m do
    write(' ',a[i]);
  writeln;
end;
```

```
procedure Back(pas:integer);
var i:integer;
begin
  if pas = m + 1 then
    ScribeSolutie
  else
    for i:=a[pas-1]+1 to n do
      begin
        a[pas]:=i;
        Back(pas+1);
      end;
end;
```

```
begin
  readln(n,m);
  a[0]:=0;
  Back(1);
end.
```

- a) determină toate partițiile numărului n formate din m termeni
 b) determină toate submulțimile mulțimii $\{1, 2, \dots, m\}$
 c) determină permutările mulțimii $\{1, 2, \dots, n\}$
 d) determină toate submulțimile mulțimii $\{1, 2, \dots, n\}$ care au m elemente

Versiune C:

Ce realizează următorul program:

```
#include <stdio.h>
int m,n,a[101];

void ScribeSolutie(void) {
  int i;
  printf("%d",a[1]);
  for (i=2;i<=m;i++)
```





```

    printf(" %d",a[i]);
    putchar('\n');
}
void Back(int pas){
    int i;
    if (pas == m + 1)
        ScribeSolutie();
    else
        for(i=a[pas-1]+1;i<=n;i++){
            a[pas] = i;
            Back(pas+1);
        }
}

```

```

void main(void){
    scanf("%d%d",&n,&m);
    a[0] = 0;
    Back(1);
}

```

- a) determină toate partițiile numărului n formate din m termeni
- b) determină toate submulțimile mulțimii $\{1, 2, \dots, m\}$
- c) determină permutările mulțimii $\{1, 2, \dots, n\}$
- d) determină toate submulțimile mulțimii $\{1, 2, \dots, n\}$ care au m elemente

Întrebarea #31:

O tablă de șah poate fi privită ca fiind o matrice cu opt linii și opt coloane. Două turnuri se atacă unul pe altul dacă se află pe aceeași linie sau pe aceeași coloană. Problema determinării tuturor posibilităților de amplasare a opt turnuri pe o tablă de șah astfel încât oricare două dintre ele să nu se atace este echivalentă cu:

- a) problema celor opt regine (dame)
- b) problema determinării tuturor permutărilor mulțimii $\{1, 2, 3, 4, 5, 6, 7, 8\}$
- c) problema determinării tuturor submulțimilor mulțimii $\{1, 2, 3, 4, 5, 6, 7, 8\}$
- d) nici una dintre cele trei

Întrebarea #32:

Versiune Pascal:

Următorul program ar trebui să determine toate submulțimile mulțimii $\{1, 2, \dots, n\}$. El conține o singură eroare. Precizați numărul liniei pe care se află eroarea.

```

1: program Submultimi;
2: var a:array[0..100] of integer;
3:     n:integer;

4: procedure ScribeSolutie(pas:integer);
5: var i:integer;
6: begin
7:     write(' ');
8:     if pas>=1 then
9:         write(a[1]);
10:    for i:=2 to pas do
11:        write(' ',a[i]);
12:    writeln('}')
13: end;

14: procedure Sub(pas:integer);
15: var i:integer;
16: begin
17:     ScribeSolutie(pas-1);

```

```

18: for i:=a[pas-1]+1 to n do
19: begin
20:     a[pas]:=i;
21:     if pas<n then
22:         Sub(pas+1)
23:     end
24: end;

25: begin
26:     readln(n);
27:     a[0]:=0;
28:     Sub(1)
29: end.

```

- a) 8 b) 17 c) 21 d) 22

Versiune C:

Următorul program ar trebui să determine toate submulțimile mulțimii $\{1, 2, \dots, n\}$. El conține o singură eroare. Precizați numărul liniei pe care se află eroarea.

```

1: #include <stdio.h>
2: int n,a[101];

3: void ScribeSolutie(int pas){
4:     int i;
5:     putchar('{');
6:     if (pas>=1)
7:         printf("%d",a[1]);
8:     for (i=2;i<=pas;i++)
9:         printf(" %d",a[i]);
10:    puts("}");
11: }

12: void Back(int pas){
13:     int i;
14:     ScribeSolutie(pas-1);
15:     for (i=a[pas-1]+1;i<=n;i++){
16:         a[pas] = i;
17:         if (pas < n)
18:             Back(pas+1);
19:     }
20: }

21: void main(void){
22:     scanf("%d",&n);
23:     a[0] = 0;
24:     Back(1);
25: }

```

- a) 6 b) 14 c) 17 d) 18

Întrebarea #33:

Versiune Pascal:

Ce se va afișa pe ecran în urma executării următorului program?

```

program Backtracking;
var a:array[0..100] of integer;
    n:integer;

procedure ScribeSolutie;
var i:integer;
begin
    write(a[1]);
    for i:=2 to n do
        write(' ',a[i]);
    writeln;
end;

```

```

procedure Back(pas:integer);
var i:integer;
begin
    for i:=1 to n do
        begin
            a[pas]:=i;
            if pas>0 then
                Back(pas-1)
            else
                if pas = n + 1 then
                    ScribeSolutie;
            end
        end;
begin
    readln(n);
    a[0]:=0;
    Back(n)
end.

```

- a) nimic
- b) permutările unei mulțimi
- c) submulțimile unei mulțimi
- d) elementele unui produs cartezian

Versiune C:

Ce se va afișa pe ecran în urma executării următorului program?

```

#include <stdio.h>
int n,a[101];

void ScribeSolutie(){
    int i;
    printf("%d",a[1]);
    for (i=2;i<=n;i++){
        printf(" %d",a[i]);
        putchar('\n');
    }
}

void Back(int pas){
    int i;
    for (i=1;i<=n;i++){
        a[pas] = i;
        if (pas > 0)
            Back(pas-1);
        else
            if (pas == n + 1)
                ScribeSolutie();
    }
}

void main(void){
    scanf("%d",&n);
    a[0] = 0;
    Back(n);
}

```

- a) nimic
- b) permutările unei mulțimi
- c) submulțimile unei mulțimi
- d) elementele unui produs cartezian

Întrebarea #34:

Se consideră o mulțime al cărei cardinal este 6. Aceasta va avea a permutări, b submulțimi, dintre care c submulțimi vor avea cel puțin 3 elemente. Valorile a, b și c sunt:

- a) a = 720, b = 32, c = 16
- b) a = 720, b = 64, c = 42

- c) a = 720, b = 64, c = 32
- d) a = 120, b = 32, c = 16

Întrebarea #35:

Care dintre următoarele afirmații este falsă:

- a) din stive elementele sunt eliminate în ordinea inversă inserării lor
- b) din cozi elementele sunt eliminate în ordinea inserării lor
- c) stivele pot fi utilizate pentru a simula recursivitatea directă
- d) cozile pot fi utilizate pentru a simula recursivitatea indirectă

Întrebarea #36:

Pentru a insera un element într-o listă dublu înlănțuită ordonată trebuie modificată(e):

- a) două referințe
- b) patru referințe
- c) o referință
- d) nici o referință.

Întrebarea #37:

Se consideră o stivă inițial vidă. Care este suma valorilor afișate în urma execuției următoarelor operații:

```

Adaugă_în_stivă(1);
Adaugă_în_stivă(2);
Elimină_primul_element_al_stivei;
Adaugă_în_stivă(3);
Adaugă_în_stivă(4);
Scrie_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Adaugă_în_stivă(5);
Adaugă_în_stivă(6);
Elimină_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Adaugă_în_stivă(7);
Adaugă_în_stivă(8);
Scrie_primul_element_al_stivei;
Adaugă_în_stivă(9);
Elimină_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Adaugă_în_stivă(10);
Elimină_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Elimină_primul_element_al_stivei;
Scrie_primul_element_al_stivei;
Elimină_primul_element_al_stivei;

```

- a) 35 b) 37 c) 39 d) 41

Întrebarea #38:

Ce se înțelege prin santinelă, în cazul structurilor dinamice de date?

- a) un element fictiv care nu face parte din structura propriu-zisă și care este utilizat pentru simplificarea implementării operațiilor de inserare și eliminare
- b) un element al structurii care conține primul element din structură potrivit unui anumit criteriu de ordonare





- c) primul element al structurii
d) nici una dintre cele trei

Întrebarea #39:

Care dintre următoarele afirmații este falsă?

- a) un graf neorientat trebuie să conțină cel puțin o muchie
b) un graf neorientat poate avea mai multe vârfuri izolate
c) o muchie poate fi privită ca fiind o pereche de noduri
d) lungimea unui ciclu este dată de numărul muchiilor care formează ciclul respectiv

Întrebarea #40:

Care dintre următoarele afirmații este adevărată?

- a) un subgraf al unui graf G are întotdeauna același număr de noduri ca și graful G
b) un graf conex este întotdeauna și un graf eulerian
c) un graf complet este întotdeauna și un graf regulat
d) un graf hamiltonian este întotdeauna și un graf regulat

Întrebarea #41:

Care este gradul maxim al unui nod din graful neorientat descris prin următoarea matrice de adiacență?

```

0 1 0 1 0 0 0 1 1 1
1 0 1 1 1 1 1 0 1 1
0 1 0 0 1 0 0 1 1 0
1 1 0 0 0 1 0 1 1 0
0 1 1 0 0 1 1 1 0 1
0 1 0 1 1 0 1 0 0 0
0 1 0 0 1 1 0 0 0 1
1 0 1 1 1 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0
1 1 0 0 1 0 1 0 0 0

```

- a) 6 b) 7 c) 8 d) 9

Întrebarea #42:

Într-un graf orientat, gradul unui nod este dat de:

- a) maximul dintre gradul interior și gradul exterior
b) minimul dintre gradul interior și gradul exterior
c) suma dintre gradul interior și gradul exterior
d) nici una dintre cele trei

Întrebarea #43:

Care dintre următoarele corespondențe nu este corectă:

- a) noțiunii de muchie în grafurile neorientate îi corespunde noțiunea de arc în grafurile orientate
b) noțiunii de ciclu în grafurile neorientate îi corespunde noțiunea de circuit în grafurile orientate
c) noțiunii de lanț în grafurile neorientate îi corespunde noțiunea de drum în grafurile orientate
d) noțiunii de adiacență în grafurile neorientate îi corespunde noțiunea de incidență în grafurile orientate

Întrebarea #44:

Care dintre următoarele afirmații este falsă?

- a) o frunză într-un arbore cu rădăcină are întotdeauna gradul 1
b) singurul nod dintr-un arbore cu rădăcină care nu are părinte este rădăcina
c) orice nod al unui arbore cu rădăcină poate avea cel mult un frate
d) nici una dintre cele trei

Întrebarea #45:

Se consideră un arbore a cărui rădăcină este nodul 1 și care conține alte 9 noduri identificate prin numere cuprinse între 2 și 10. Arborele conține muchii între următoarele perechi de noduri: (1, 2), (1, 5), (2, 7), (3, 5), (3, 10), (4, 5), (6, 10), (7, 9) și (8, 10). Stabiliți șirul părinților celor nouă noduri (identificate prin numere cuprinse între 2 și 10), în ordinea dată de numerele de identificare ale acestor noduri. Așadar, primul element al șirului va indica numărul de ordine al părintelui nodului 2, al doilea element va indica numărul de ordine al părintelui nodului 3 etc.

- a) 1 5 5 1 10 2 10 7 3
b) 1 5 5 3 10 2 10 7 6
c) 1 2 3 4 5 6 7 8 9
d) 9 8 7 6 5 4 3 2 1

Soluții

Grila completată corect pentru acest test, împreună cu explicațiile referitoare la răspunsurile corecte, va fi publicată în următorul număr al revistei noastre.

Cei care doresc să afle mai repede răspunsurile corecte, pot să decupeze grila alăturată completată cu răspunsurile și să o trimită pe adresa redacției. În scrisoare trebuie indicată adresa *e-mail* la care se dorește primirea răspunsurilor corecte.

Observații

Caracterul acestui test este orientativ. Acest test este neoficial, el ilustrând doar concepția noastră referitoare la modul în care ar putea să arate o astfel de grilă. Este foarte posibil ca grila de la *Bacalaureat* să fie complet diferită.

Redacția *GInfo* nu își asumă nici o răspundere privitoare la relevanța și corectitudinea acestui test.

Grila constă în 45 de întrebări, pentru fiecare existând patru variante de răspuns dintre care *exact una* este corectă.

Grila

- | | | |
|---|---|---|
| 1. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 16. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 31. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 2. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 17. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 32. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 3. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 18. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 33. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 4. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 19. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 34. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 5. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 20. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 35. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 6. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 21. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 36. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 7. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 22. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 37. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 8. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 23. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 38. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 9. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 24. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 39. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 10. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 25. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 40. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 11. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 26. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 41. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 12. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 27. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 42. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 13. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 28. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 43. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 14. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 29. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 44. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |
| 15. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 30. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> | 45. a. <input type="checkbox"/> b. <input type="checkbox"/> c. <input type="checkbox"/> d. <input type="checkbox"/> |