Culegere de INFORMATICĂ

Pentru admitere la Facultatea de Matematică și Informatică a Universității din București

Lixandru Andrei-Stefan
Alexandru Mihai
Ariana Ionescu

Alex Cernat
Florin Vasiliu
Alexandru Mihai

Marian Dimofte
Adrian Rotarasu
Mihail Mihai

Profesor îndrumător:Anca Dobrovat

București
2022

Cuprins

1. Variabila x este de tip real. Pentru a verifica dacă valoarea variabilei x aparține mulțimii $[-5, 12] \cap (3, 20)$ se va utiliza următoarea expresie:

a) $(x \leq 12)\ \&\&\ (x > 3)$
b) $(x \leq 12)\ ||(x > 3)$
c) $(x < 20)\ ||\ (x \geq -5)$
d) $(x \leq 12)\ \&\&\ (x \geq 3)$

2. Variabila x este de tip real. Pentru a verifica dacă valoarea variabilei x aparține mulțimii $[-15, 12] \cup \{13, 14, 15\}$ se va utiliza următoarea expresie:

a) $!\ ((x \geq -15)\ ||\ (x \leq 12)\ \&\&\ (x != 13)\ \&\&\ (x != 14) \&\& (x != 15))$
b) $!\ (!\ ((x \geq -15)\ \&\&\ (x \leq 12))\ \&\&\ (x == 13)\ \&\&\ (x == 14) \&\& (x == 15))$
c) $!\ ((x <- 15)\ ||\ (x > 12)\ \&\&\ (x != 13)\ \&\&\ (x != 14) \&\& (x == 15))$
d) $!\ ((x <- 15)\ ||\ (x > 12)\ \&\&\ (x != 13)\ \&\&\ (x != 14) \&\& (x != 15))$

3. Variabilele x şi y sunt tip int. Care dintre expresiile C/C++ de mai jos are valoarea 1 dacă şi numai dacă valorile întregi nenule memorate în variabilele x şi y sunt egale?

a) $(x\%y == 0) \&\& (y\%x == 0) \&\& (x * y > 0)$
b) $!\ ((x \leq y) \&\& (y > x))$
c) $(x \leq y)\ ||\ (y \leq x)$
d) $pow(x, 2) == y * y$

4. Care dintre expresiile urmatoare este echivalentă cu expresia:

$$((a > 3) \&\& (a < 15))\ ||\ (a != b)?$$

a) $!\ ((a \leq 3)\ ||\ (a > 15))\ ||\ !\ (a == b)$
b) $!\ ((a \leq 3)\ ||\ (a \geq 15))\ ||\ !\ (a == b)$
c) $!\ ((a < 3)\ ||\ (a \geq 15))\ ||\ !\ ((a == b)$
d) $!\ (!\ ((a > 3)\ ||\ (a < 15))\ ||\ !\ ((a == b)$

5. Variabilele x,y şi t sunt reale.Știind că x < y, care din următoarele expresii are valoarea 1 dacă și numai dacă numărul memorat în t **NU** aparține intervalului (x, y)?

a) $!\ ((t > x) \&\& (t \leq y))$
b) $!\ ((t \geq x)\ ||\ (t \leq y))$
c) $!\ ((t > x)\ ||\ (t < y))$
d) $!\ ((t > x)\ \&\&\ (t < y))$

6. Precizați ce se va afișa după executarea secvenței de program de mai jos

$a[0] = 3; a[1] = 4; a[2] = 5;$

for($i = 0; i < 3 \&\& aux == 0; i ++$)
if(pow(a[i%3],2) == pow(a[(i+1)%3],2) + pow(a[(i+2)%3],2)
aux = 1;
cout<<aux;

a)nu compilează
b)1
c)0
d)"aux"

7. Fie a[0]; a[1]; a[2] laturile unui triunghi oarecare.Pentru ce tip de triunghi secvența de mai jos afisează 1?

$a[0] = 3; a[1] = 4; a[2] = 5;$

for($i = 0; i < 3 \&\& aux == 0; i ++$)
if(pow(a[i%3],2) == pow(a[(i+1)%3],2) + pow(a[(i+2)%3],2)
aux = 1;
cout<<aux;

a)nu compilează
b)triunghi oarecare
c)triunghi dreptunghic
d)triunghi isoscel

8. Fie n un numar natural .

for($i = 1$;i$\leq n$;i ++)
for(j = n;$j \geq i$;j −−)
for($k = i; k \leq j; k$ ++)
cout<<"*";

Precizați care dintre următoarele afirmații este adevărată:

a) Pentru n = 3 subalgoritmul afișează 9 steluțe
b) Pentru n = 5 subalgoritmul afișează 10 steluțe
c) Pentru n = 3 subalgoritmul afișează 11 steluțe
d) Pentru n = 5 subalgoritmul afișează 35 steluțe

9. Fie [x] partea întreagă a numărului x .Indicați o expresie C/C++ care afișează același rezultat ca a expresiei aritmetice alăturate: $\frac{([7,13]+1)}{2+[6,14]}$

a) $(floor(7.13)/(2 + ceil(6.14))$
b) $((floor(7.13) + 1)/(2 + ceil(6.14))$
c) $ceil(7.13)/(2 + ceil(6.14))$
d) $ceil(7.13)/(2 + floor(6.14))$

10. Oare ce face?

$$int\ main(){$$

$$int\ n, k, aux, i, numarator\_1\ =\ 1, numarator\_1\ =\ 1, numitor\_2\ =\ 1;$$

$$cin\gg n;$$

$$aux\ =\ 1;$$

$$cout\ll"1\ \backslash n";$$

$$while(aux\ <=\ n){$$

$$for(k\ =\ 0; k\ <=\ aux; k\ ++)$$

$$if(k\ ==\ aux\ ||\ k\ ==\ 0){$$

$$cout\ll"1";$$

$$if(k\ ==\ aux)\ cout\ll endl;}$$

$$else\{$$

$$for(i\ =\ 1; i\ <\ aux; i\ ++)$$

$$numarator\_1\ =\ numarator\_1\ *\ i;$$

$$for(i\ =\ 1; i\ <=\ (aux\ -\ 1)\ -\ (k\ -\ 1); i\ ++)$$

$$numitor\_1\ =\ numitor\_1\ *\ i;$$

$$for(i\ =\ 2; i\ <\ k; i\ ++)$$

$$numitor\_1\ =\ numitor\_1\ *\ i;$$

$$for(i\ =\ 1; i\leq(aux\ -\ 1)\ -\ k; i\ ++)$$

$$numitor\_2\ =\ numitor\_2\ *\ i;$$

$$for(i\ =\ 1; i\ <=\ k; i\ ++)$$

$$numitor\_2\ =\ numitor\_2\ *\ i;$$

$$cout\ll(numarator\_1/numitor\_1)\ +\ (numarator\_1/numitor\_2)\ll"\ ";$$

$$numarator\_1\ =\ 1; numitor\_1\ =\ 1; numitor\_2\ =\ 1;\}$$

$$aux\ ++;$$

$$return\ 0;\}$$

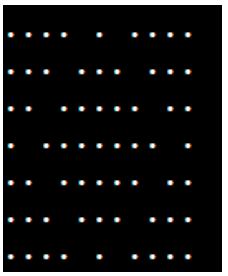
a) Afișează triunghiul lui Pascal
b) Nu compilează
c) Calculează suma $C_n^0 + C_n^1 + \ ... \ + C_n^n$
d) Nu funcționează pentru n= $2k\ +\ 1$,∀k∈N

11. Pentru ce valori secventa urmatoare afiseaza un numar intreg?

$$cout \ll sqrt(abs(pow(b, 2) - 4 * a * c))/(2 * a)$$

a) $(1, 2, 3)$
b) $(1, 2, 9)$
c) $(1, 2, 10)$
d) $(1, 3, 19)$

12. Cu ce putem înlocui punctele de suspensie roșii pentru a se afișa imaginea alaturata:



$$const\ char\ *\ a\ =\ "....";$$

$$int\ i, j\ =\ 0, k;$$

$$k\ =\ strlen(a);$$

$$for(i\ =\ 0; i\ <\ strlen(a); i\ ++)\{$$

$$cout \ll ... \ll\ " "\ll a\ +\ strlen(a) - i\ +\ 1 \ll a\ +\ k \ll " "\ \ll ...;$$

$$k\ --;$$

$$if(i! =\ strlen(a) - 1)\{cout \ll endl;\}$$

$$\}$$

$$cout \ll endl;$$

$$k\ =\ 0;$$

$$for(i\ =\ strlen(a); i \geq 0; i\ ++)\{$$

$$if(k! =\ 0)$$

$$cout \ll a\ +\ i \ll " "\ll a\ +\ j\ -\ 1 \ll a\ +\ k \ll " "\ll a\ +\ i \ll endl;$$

$$k\ ++;$$

$$j\ ++;\}$$

a) $a\ +\ i\ -\ 1$
b) $a\ +\ i\ +\ 1$

c) $a + i$
d) $a - i$

Grafuri si arbori cu rădăcină

1. Parcurgerile în postordine şi preordine ale unui arbore binar sunt d e b f g c a şi respectiv a b d e c f g. Parcurgerea în inordine a aceluiaşi arbore este:

a) f g c a b d e
b) d b e a f c g
c) a b c d e f g
d) g f c a e d b

2. Se consideră graful neorientat definit prin mulţimea vârfurilor V = {1, 2, 3, 4, 5, 6} si multimea muchiilor E = { [1, 2], [1, 4], [2, 3], [2, 6], [3, 6], [4, 3], [4, 5] }. Ordinea în care sunt vizitate vârfurile corespunzător parcurgerii in adancime (DFS) pornind din varful 4 poate fi:

a) 4, 1, 3, 5, 2, 6

b) 4, 1, 5, 2, 3, 6

c) 4, 1, 2, 3, 5, 6

d) 4, 1, 2, 3, 6, 5

3. Precizati care este numărul maxim de noduri de grad 3 într-un graf neorientat cu 5 noduri?

a) 2

b) 3

c) 4

d) 5

4. Se consideră un graf neorientat G cu 12 noduri si 7 muchii. Care este numărul maxim de componente conexe din care poate fi format graful G?

a) 5

b) 6

c) 7

d) 8

5. Se consideră un graf conex neorientat G cu 39 muchii. Care este numărul total de noduri ale grafului, știind că 7 dintre noduri au gradul 2, 2 nodurii gradul 5 și restul au gradul 6?

a) 11

b) 14

c) 18

d) 19

6. Se consideră un graf neorientat cu 5 noduri numerotate de la 1 la 8 și următoarele muchii: [1,2], [1,3], [1,4], [2,3], [2,5], [3,4], [3,5]. Precizați care este numărul minim de culori cu care pot fi colorate nodurile grafului, astfel încât oricare două noduri adiacente să aibă culori diferite.

a) 2

b) 3

c) 4

d) 5

7. Se consideră un graf bipartit cu n noduri. Precizați care este numărul minim de culori cu care pot fi colorate nodurile grafului, astfel încât oricare două noduri adiacente să aibă culori diferite.

a) 1

b) 2

c) 3

d) n

8. Fie T un arbore cu rădăcină. Arborele are 8 noduri numerotate de la 1 la 8 şi este descris prin următorul vector „de taţi”: (4,5,0,3,4,5,4,5). Care sunt frunzele arborelui?

a) 2, 6, 8

b) 1, 2, 6, 7, 8

c) 1, 7

d) 1, 5, 7

9. Câte frunze are arborele cu rădăcină cu 9 noduri, numerotate de la 1 la 9, având următorul vector de „taţi”: t=(9,3,4,7,3,9,0,7,2)?

a) 2

b) 3

c) 4

d) 5

10. Care este înălţimea unui arbore cu rădăcină cu 11 noduri, numerotate de la 1 la 11, având următorul vector de „taţi”: t= (6,5,5,2,0,3,3,3,8, 7, 7). Prin înălţimea unui arbore cu rădăcină înţelegem lungimea maximă a unui lanț de la rădăcină la un nod al arborelui.

a) 2

b) 3

c) 4

d) 5

11. Care este numărul minim de noduri dintr-un arbore binar complet cu înălţimea 3? Prin înălţimea unui arbore înţelegem lungimea maximă a unui lanț de la rădăcină la un nod al arborelui.

a) 3

b) 4

c) 8

d) 15

12. Fie T un arbore cu rădăcină. Arborele are 9 noduri numerotate de la 1 la 9 şi este descris prin următorul vector „de taţi”: (4,5,4,5,0,2,8,2, 6). Ordinea în care sunt vizitate nodurile corespunzător parcurgerii in preordine este:

a) 5 4 2 1 3 6 8 9 7

b) 5 4 1 3 2 6 9 8 7

c) 5 2 8 7 4 3 1 6 9

d) 5 2 8 7 6 9 4 3 1

13. Fie T un arbore cu rădăcină. Arborele are 9 noduri numerotate de la 1 la 9 şi este descris prin următorul vector „de taţi”: (4,5,4,5,0,2,8,2, 6). Ordinea în care sunt vizitate nodurile corespunzător parcurgerii în postordine este:

a) 1 4 3 9 7 6 8 2 5

b) 1 3 4 9 6 7 8 2 5

c) 1 3 4 9 7 6 8 2 5

d) 1 4 3 9 6 7 8 2 5

14. Care este înălţimea minima a unui arbore binar cu 14 noduri. Prin înălţimea unui arbore cu rădăcină înțelegem lungimea maximă a unui lanț de la rădăcină la un nod al arborelui.

a) 2

b) 3

c) 4

d) 5

15. Se consideră un graf neorientat G cu 100 de noduri. Care este numărul maxim de muchii pe care le poate avea graful, pentru ca acesta sa nu fie conex?

a) 2451

b) 4850

c) 9801

d) 4851

16. Fie un graf neorientat, complet, cu 17 noduri. Câte muchii are acesta?

a) 136
b) 148
c) 153
d) 141

17. Fie un graf neorientat cu 10 noduri, numerotate de la 1 la 10 şi muchiile [1,2], [1,5],[1,8], [1,10], [3,7], [3,8], [3,9], [5,6], [6,7], [7,9], [9,10]. Nodurile ce au gradul divizibil cu 3 sunt:

a) 3, 7, 9
b) 2
c) 3, 4, 7, 9
d) 4

18. Fie un graf neorientat cu 60 de noduri şi 100 de muchii. Numărul maxim de componente conexe pe care îl poate avea acesta este:

a) 47
b) 46
c) 3
d) 4

19. Fie un arbore cu 12 noduri reprezentat prin vectorul de tați [10, 6 , 7, 7, 3, 4, 0, 4, 3,12, 10, 7]. Descendenții nodului 4 sunt:

a) 6, 8
b) 3, 12
c) 2, 3, 6
d) 2, 6, 8

20. Înălțimea minimă a unui arbore cu 1273 de noduri, în care fiecare nod are maxim 3 fiieste:

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

21. Fie un graf neorientat cu 7 noduri şi cu muchiile [1,2], [1,3], [2,3], [2,6], [3,4], [4,7], [4,5], [5,6], [6,7], [7,2]. Numărul de cicluri al acestui graf este:

a) 1
b) 2
c) 12
d) 3

22. Numărul grafurilor orientate, complete este:

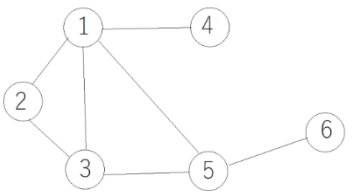
a) 34
b) 310
c) 36
d) 312

23. Un graf conex cu toate gradele pare se numește:

a) bipartit
b) eulerian
c) hamiltonian
d) complet

24. Dacă din graful alaturat eliminăm vârful 5 am obtinut un:

a) subgraf



b) graf partial
c) graf initial
d) graf complementar

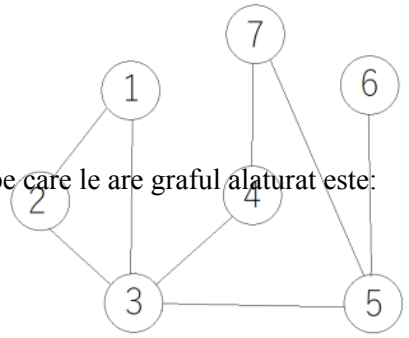
25. Numărul de subgrafuri pe care le admite un graf cu 7 vârfuri şi 8 muchii este:

a) 128
b) 127
c) 256
d) 255

26. Fie un graf neorientat, ciclic, cu 10 noduri. Numărul de muchii ce ar trebui adăugate pentru ca graful să devină eulerian este:

a) 9
b) 0
c) 1
d) 8

27. Numărul de perechi de muchii adiacente pe care le are graful alaturat este:



a) 11
b) 14
c) 12
d) 13

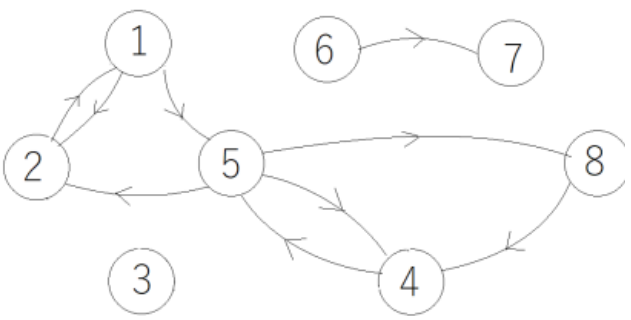
28. Fie t un vector de tati. Cu ce trebuie înlocuite punctele de suspensie din programul alaturat, astfel încât acesta să afişeze câte frunze există în arborele descris de t?

a) $for(j = 1; j <= 8; j ++)\, if(i == j)\, ok = 0;$
b) $for(j = 1; j <= 8; j ++)\, if(t[i] == t[j])\, ok = 0;$
c) $for(j = 1; j <= 8; j ++)\, if(i == t[j])\, ok = 0;$
d) $for(j = 1; j < 7; j ++)\, if(t[i] == j)\, ok = 0;$

```
#include <iostream>
using namespace std;
int main()
{
    int i,j,nr_frunze=0,ok;
    int t[10]={2,0,2,3,3,1,1};
    for (i=1;i<=7;i++)
    {   ok=1;
        ...
        if(ok)
            nr_frunze++;
    }
    cout <<"Numarul de frunze
este: "<<nr_frunze;
    return 0;
}
```

29. Numărul de componente tare conexe ale grafului orientat alaturat este:

a) 5
b) 6
c) 3
d) 4



30. Care dintre următoarele afirmații sunt corecte
    1. Matricea de adiacență a unui graf orientat are întotdeauna 0 pe diagonala principală.
    2. Un graf în care fiecare nod are gradul mai mare sau egal cu jumătatea numărului de noduri este un graf eulerian.
    3. Un graf neorientat, complet cu n noduri are n(n+1)/2 muchii.
    4. Un graf bipartit are 2 elemente conexe.
    5. Un graf orientat este echilibrat dacă gradul extern este egal cu gradul intern pentru fiecare nod.

    a) 1, 3, 5
    b) 1, 4
    c) 2, 5, 4
    d) 1, 5

31. Fie graful G = (V, E) cu V = {1, 2, 3, 4, 5, 6}, E = {(1,2), (1,4), (2,4), (4,5),(5,6)} si v0 = 2.
    Ordinea in care sunt vizitate varfurile corespunzator parcurgerii in latime BF este:
    a) 2, 1, 4, 5, 6
    b) 2, 3, 1, 5, 6, 4
    c) 1, 2, 4, 5, 6
    d) 2, 1, 4, 5, 3, 6

32. Intr-un graf neorientat G, notam cu n nr de varfuri si cu m nr de muchii. Daca graful este
    un arbore atunci intre n si m exista urmatoarea relatie matematica:
    a) $n = m - 1$
    b) $m = n + 2$
    c) $n = m + 1$
    d) $m = n$

33. Fie graful G = (V, E) graf, cu V = (1, 2, 3, 4, 5, 6, 7, 8, 9), E = ((1, 2), (1, 4), (2, 7), (2, 8),(3, 6), (3, 9), (4, 5), (4, 7), (7, 8)) si v0 = 4. Ordinea in care sunt vizitate varfurile corespunzator parcurgerii in adancime DF este:
    a) $3, 6, 9$
    b) $4, 2, 1, 5, 7, 8$
    c) $4, 1, 2, 7, 8, 5, 3$
    d) $4, 1, 2, 7, 8, 5$

34. Un graf neorientat G contine un arbore partial daca si numai daca G este:
    a) $aciclic$
    b) $eulerian$
    c) $hamiltonian$

d) $conex$

35. Fie graful G=(V,E) cu V={1,2,3,4,5} si E={(1,2);(1,4);(2,3);(2,5);(3,4);(3,5);(4;5)}.Determinati care este nr. Minim de muchii care pot fi eliminate din graf astfel incat in graful partial rezultat sa existe exact un vf de grad 0?
    a) 3
    b) 2
    c) 5
    d) 1

36. Fie graful G=(V,E) , cu V={A,B,C,D,E,F} si E={(A,E), (B,D), (B,E), (C,D), (E,F)}. Care este nodul ce poate fi radacina a arborelui a.i. fiecare nod care nu este de tip frunza sa aiba un nr. Impar de descendenti directi(fii)?

a) C
b) A
c) D
d) E

37. Fie graful G=(V,E) , cu V={A,B,C,D,E,F} si E={(A,B), (B,E), (B,F), (C,E), (C,D)}. Care este nodul ce poate fi ales ca radacina a arborelui a.i. fiecare nod care nu este de tip frunza sa aiba un nr. Impar de descendenti directi(fii)?

a) A
b) B
c) C
d) D

Subprograme recursive

1. Dat fiind următorul subprogram, ce se afișează în urma apelului f(5)?

$$void\ f(int\ x)\{$$

$$cout \ll x; \quad //\ printf("\%d", x);$$

$$if(x > 0)\{$$

```
if(x%2 == 0){
   f(x/2);
cout<< * ";    // printf(" * ");
    }
  else
f(x − 1);}}
```

a) 54210**
b) 54*2*10
c) 5210**
d) 5*4*210

2. Ce returnează apelul f(71, 14)?

```
int f(int a, int b){
int s, i;
if(a == 0)
return 1;
else{
s = 0;
for(i = 2; i < b; i *= 2)
s += f(a/(i * b), b);
return s;}}
```

a) 4
b) 7
c) 12
d) 6

3. Care valoare **NU** poate fi returnată de următorul subprogram?

```
int f(int x){
if(x == 0)
return 1;
```

$$if(x < 0)$$

$$return\ 2\ *\ f(-\ x\ -\ 1);$$

$$return\ 3\ *\ f(-\ x\ +\ 1);\}$$

a) 24
b) 36
c) 18
d) 432

4. Câte caractere " * " se afișează la apelul f(7, 5)?

$$void\ f(int\ k, int\ y)\{$$

$$if(k\%3\ ==\ 0||\ y\%3\ ==\ 0)\{$$

$$cout\ll\ *\ ";\quad //\ printf("\ *\ ");$$

$$y\ --;\}$$

$$if(k > 0)$$

$$f(k\ -\ 1, y);\}$$

a) 2
b) 3
c) 4
d) 5

5. Vectorul v conține un număr par, nenul de elemente întregi, indexate de la 1 la n inclusiv.Care condiție poate înlocui punctele de suspensie astfel încât apelul $s(v, 1, n, 1)$să returneze suma elementelor vectorului?

$$int\ s(int\ v[], int\ i, int\ n, int\ b)\{$$

$$if(\ldots)$$

$$return\ v[i];$$

$$return\ v[i]+\ s(v, n\ -\ i\ +\ b, n, 3\ -\ b);\}$$

a) 2 * i == n + b
b) n + b - i - 1 == i
c) i == n
d) i == n / 2

6. Se dau subprogramele definite mai jos. Ce returnează apelul $g(12)$?

$$void\ f(int\ \&a)\{$$

$$a\ +=\ a\%2;\}$$

$$int\ g(int\ n)\{$$

$$int\ p\ =\ n/2;$$

$$if(n\ <=\ 0)$$

$$return\ 1;$$

$$f(p);$$

$$return\ p\ *\ g(n\ -\ p\ -\ 1);\}$$

a) 36
b) 12
c) 42
d) 24

7. Care variantă **NU** poate fi afișată prin rularea subprogramului f?

$$void\ f(int\ n)\{$$

$$if(n\%5\ ==\ 0)$$

$$cout\ll"c";\quad //\ printf("c");$$

$$if(n\ >\ 0)\{$$

$$f(n/2);$$

$$cout\ll n\%10;\quad //\ printf("\%d", n\%10);$$

$$\}\}$$

a) ccc125
b) cc1375
c) ccc1250
d) c1363

8. Pentru câte valori ale argumentului din intervalul $[2^9, 2^{12})$returnează următorul subprogram valoarea 10?

$$int\ f(int\ n)\{$$

$$if(n\ ==\ 0)$$

$$return\ 0;$$

$$return\ n\%2\ +\ f(n/2);\}$$

a) 100
b) 32
c) 66
d) 22

9. Variabila n este declarată global și inițializată cu 1. Fie subprogramele:

$$int\ f(int\ \&a)\{$$

$$if(a\%3!=\ 0)$$

$$a\ ++;$$

$$return\ 2\ *\ a;\}$$

$$void\ g(int\ n)\{$$

$$cout\ll f(n);\quad //\ printf("\%d", n);$$

$$\}$$

Ce afișează următoarea secvență?

$$cout\ll f(n)\quad //printf("\%d", f(n));$$

$$g(n);$$

$$cout\ll n;\quad //\ printf("\%d", n);$$

a) 442
b) 443
c) 463
d) 462

10. Pentru un număr real x,câte înmulțiri se efectuează pentru apelul $f(n, p)$, $cu\ 1\leq p\leq n$?

$$int\ f(int\ n, int\ i)\{$$

$$if(i >= n)$$

$$return\ i;$$

$$return\ f(n, 2\ *\ i);\}$$

a) ceil(log2(n/p))
b) floor(log2(n/p))
c) floor(log2(n/p)) + 1
d) ceil(log2(n/p)) – 1

11. Care este suma cifrelor afișate de următorul subprogram la apelul f(70, 80)?

$$void\ f(int\ x, int\ y)\{$$

$$if(x < y)$$

$$cout\ll"0";\quad //\ printf("0");$$

$$else\{$$

$$x = y - 1;$$

$$cout\ll"1";\ //\ printf("1");$$

$$\}$$

$$if(y > 0)\{$$

$$f(y, x);$$

$$cout\ll"2"; \quad // \quad printf("2");$$

$$\}\}$$

a) 211
b) 213
c) 215
d) 210

12. Ce afișează următorul subprogram la apelul f(7)?

$$int\ f(int\ n)\{$$

$$int\ x\ =\ 1;$$

$$cout\ll n\%2\ll" "; \quad // \quad printf("\%d",\ n\%2);$$

$$if(n\ >\ 0)\{$$

$$x\ =\ f(n/2);$$

$$cout\ll x\%2\ll" "; \quad // \quad printf("\%d", x\%2);$$

$$\}$$

$$return\ 2\ *\ x;\}$$

a) 1 0 1 0 1 0 0
b) 1 1 1 0 0 0
c) 1 1 1 0 0 0 0
d) 1 1 1 1 0 0 0 0

13. Fie următorul subprogram cu „cnt" declarat global și inițializat cu 0. Parametrul actual pentru v va avea n (n > 0) elemente inițializate cu 0,indexate de la 0. Pentru care valori ale argumentelor n, p **NU** se încheie execuția apelului $f(v, n, p, 0)$?

$$void\ f(int\ v[], int\ n, int\ p, int\ i)\{$$

$$if(v[i]\ ==\ 0)\{$$

$$v[i]\ =\ 1;$$

$$cnt\ ++;\}$$

$$if(cnt\ <\ n)$$

$$f(v, n, p, (i\ +\ p)\%n);\}$$

a) 8 10
b) 5 2
c) 3 8

d) 7 15

14. Ce afiseaza urmatorul subprogram daca se citeste initial valoarea 10 pentru n?

$$void\ subprogram(int\ \&n)\{$$
$$if(n > 5)\{$$
$$n = n - 1;$$
$$subprogram(n);$$
$$cout\ll n; \quad // \ printf("\%d", n);$$
$$\}\}$$

a) 55555
b) 56789
c) 12345
d) Eroare

15. Ce afiseaza urmatorul subprogram?

$$void\ subprogram(int\ \&n)\{$$
$$if(n > 5)\{$$
$$n = n - 1;$$
$$subprogram(n);$$
$$cout\ll n;\}\}$$
$$int\ main()\{$$
$$int\ n = 10;$$
$$subprogram(n);\}$$

a) 55555
b) 56789
c) 12345
d) 5555

16. Ce afiseaza urmatorul subprogram?

$$int\ n;$$
$$void\ subprogram()\{$$
$$cout\ll n\ll " ";$$

```
int n = 100;
cout<<n<<" ";
{int n = 20;
cout<<n<<" ";}
cout<<n;}
int main(){
subprogram();}
```

a) 0 100 20 10
b) 0 100 20 100
c) 0 10 20 100
d) *Nu compilează*

17. Ce afiseaza urmatorul program?

```
void f(int x){ x = 10;}
int main(){
int x = 7;
f(x);
cout<<x;
return 0;}
```

a) 0
b) 7
c) 10
d) *Nu compilează*

18. Ce afiseaza urmatorul program?

```
int x, y;
int f(){
x ++;
return 1;}
int g(){
y ++;
return 2;}
int main(){
if(f() ||g())
cout<<x<<y;}
```

a) 00

b) 10
c) 11
d) 12

19. Ce calculeaza subprogramul?

$$int\ cauta(float\ v[], int\ n, float\ val)\{$$

$$int\ rez;$$

$$if(n == 0)\ rez =- 1;$$

$$else\ if(v[n - 1] == val)\ rez = n - 1;$$

$$else\ rez = cauta(v, n - 1, val);$$

$$return\ rez;\}$$

a) prima aparitie a unei valori date (val) intr-un vector
b) ultima aparitie a unei valori date (val) intr-un vector
c) prima si ultima aparitie a unei valori date (val) intr-un vector
d) numarul de aparitii ale unei valori date (val) intr-un vector e. prima valoare diferita de valoarea data (val)

20. Ce returneaza programul urmator daca se apeleaza s(3)?

$$int\ s(int\ n)\{$$

$$int\ rez;$$

$$if(n == 0)\ rez = 0;$$

$$else\ rez = n + s(n - 1);$$

$$return\ rez;\}$$

a) 11
b) 10
c) 7
d) 6

21. Ce returneaza functia urmatoare daca se apeleaza calc(3)?

$$int\ calc(int\ n)\{$$

$$int\ rez;$$

$$if(n == 0\ ||\ n == 1)\ rez = 1;$$

$$else\ rez = 2\ *\ calc(n - 1) + calc(n - 2);$$

$$return\ rez;\}$$

a) 21

b) 17
c) 7
d) 9

22. Ce afiseaza programul?

$$int\ x;$$
$$void\ F(){$$
$$cout\ll x;$$
$$int\ x\ =\ 10;$$
$$cout\ll x;$$
$$\{int\ x\ =\ 20;$$
$$cout\ll x;\}$$
$$cout\ll x;\}$$
$$int\ main(){$$
$$x\ =\ 5;$$
$$F();$$
$$cout\ll x;\}$$

a) 51020105
b) 51020205
c) 51020202
d) 51020100

Structuri de Date

---

1. Precizați care din următoarele șiruri de numere poate reprezenta gradele vârfurilor unui arbore cu n noduri (n număr natural cunoscut).

a) 2 1 1 1
b) 1 1 1 2 1 6
c) 4 1 1 2 1 1
d) 2 1 1 1

2. Precizați care este numărul maxim de frunze ale unui arbore binar (arbore în care fiecare nod are cel mult doi fii) cu 68 de noduri.

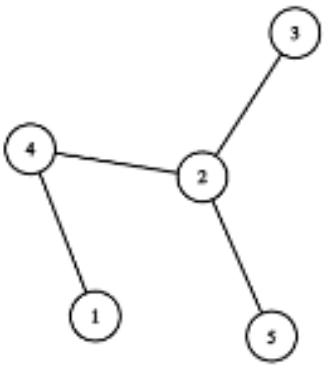
   a) 32
   b) 33
   c) 34
   d) 66

3. Numărul nodurilor terminale (frunze) ale arborelui cu rădăcină corespunzător vectorului de tați (4, 7, 0, 3, 4, 7, 3, 3) este:

a) 5
b) 4
c) 3
d) 7

4. Un arbore are nodurile numerotate cu numere de la 1 la 5. Vectorul de tați asociat arborelui poate fi:

a) 2, 1, 0, 3, 4
b) 1, 4, 0, 3, 4
c) 5, 4, 2, 1, 3
d) 4, 3, 0, 2, 2



5. Precizați care dintre următoarele matrice, este matricea de adiacenţă a unui arbore cu 4 noduri.

a)
0 1 1 1
1 0 1 0
1 1 0 0
1 0 0 0

b)
0 0 1 0
0 0 1 0
1 0 0 1
0 1 1 0

c)
0 0 1 0
0 0 0 1
1 0 0 1
0 1 1 1

d)
0 0 1 0
0 0 0 1

1 0 0 1

0 1 0 1

6. Se consideră un arbore. Referitor la un lanţ elementar care uneşte două noduri distincte a şi b, afirmaţia adevărată este:

a) Este unic, dacă şi numai dacă a sau b este frunză.
b) Nu poate trece prin rădăcina arborelui.
c) Este unic, dacă şi numai dacă a sau b este rădăcină.
d) Este unic, oricare ar fi a şi b

7. Precizați câți arbori binari cu 3 noduri, numerotate de la 1 la 3, se pot construi. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii), ordonați: fiu stâng, fiu drept. Dacă un nod are un singur descendent trebuie specificat dacă este fiu stâng sau fiu drept.

a) 9
b) 21
c) 30
d) 6

8. Un arbore binar este un arbore cu rădăcină în care fiecare nod are cel mult 2 descendenţi direcţi (fii). Un arbore binar complet, cu h niveluri, are un număr de noduri egal cu:

a) $2h - 1$
b) $2h + 1$
c) $2^h - 1$
d) $2^h + 1$

9. Un arbore oarecare cu rădăcină este reprezentat prin vectorul de taţi t. Dacă algoritmul următor determină nivelul pe care se găsește un nod x în arbore, cu ce secvență de cod se pot înlocui punctele de suspensie de mai jos?

$$int\ nivel\ =\ 0;$$

$$while(t[x])\{$$

$$....$$

$$nivel\ ++;\}$$

a) $t[x] = x + t[x]$
b) $x = t[x]$
c) $t[x] --$
d) $t[x] = x$

10. Pentru un număr natural nenul n, se construiește un arbore cu rădăcină astfel: rădăcina este numerotată n și orice nod care este numerotat cu o valoare $x > 1$ are ca fii nodurile numerotate cu divizorii săi, mai puțin numărul însuși. Toate frunzele arborelui sunt numerotate cu 1. Precizați câte dintre numerele naturale din intervalul [10, 20] pot fi alese ca rădăcină, astfel încât arborele asociat să aibă un număr maxim de frunze.

a) 1
b) 2
c) 3
d) 4

11. Precizați descendenții nodului 5 din arborele dat de următorul vector tați:
(5, 8, 4, 0, 4, 5, 3, 6, 7, 8).

a) 1,6,7
b) 4
c) 1,6
d) 1,2,6,8,10

12. Se consideră un arbore cu 8 noduri și muchiile [1, 2], [2, 3], [3, 6], [4, 3], [5, 7], [7, 2], [8, 2]. Pentru ca arborele să conțină un număr maxim de lanțuri elementare de lungime 3 care nu conțin rădăcina, se poate alege ca rădăcină oricare dintre nodurile:

a) 1,4,5,6,8
b) 1,2,4,5
c) 1,3,6,7
d) 2,3,4,7,8

13. Fie T un arbore oarecare cu un număr par de noduri, în care fiecare nod are maxim 2 fii. Numărul maxim de noduri de pe ultimul nivel i, știind că rădăcina arborelui se află pe nivelul 1, este?

a) $2^{\frac{i}{2}+1}$
b) $2^{\frac{i}{2}-1}$
c) $2^{\frac{i}{2}-1} - 1$
d) $2^{\frac{\frac{i}{2}+1}{2}}$

14. Fie arborele cu rădăcină cu nodurile numerotate de la 1 la 15, reprezentat prin vectorul de tați:
(3, 8, 5, 5, 0, 8, 3, 5, 1, 7, 7, 5, 4, 3, 6). Câți descendenți are nodul 8?

a) 5
b) 4
c) 3
d) 7

15. Fie un arbore cu rădăcină care are 2000 de noduri iar fiecare nod are maxim 4 fii. Indicați care este înălțimea minimă a arborelui?

a) 6
b) 7
c) 5
d) 4

16. Se consideră următoarea listă de descendenți asociată unui arbore cu rădăcină cu 8 noduri:

1: 4,7,6,2

2: –

3: 4,6,5,2,7,8,1

4: –

5: –

6: –

7: 2

8: 7,2,4,1,6

Vectorul de taţi asociat acestui arbore este:

a) 8 7 0 1 3 1 1 3
b) 8 7 0 1 8 1 1 3
c) 0 7 8 3 1 3 3 1
d) 8 7 8 1 3 1 1 0

17. Se consideră un arbore cu rădacină care are: 12 noduri; înălțimea 4; 6 frunze; lungimea celui mai lung lanț elementar 6; numărul de noduri de grad 1 este 7. Un posibil vector de taţi asociat acestui arbore ar putea fi:

a) 7 6 5 7 0 1 5 3 7 3 8 11
b) 12 4 6 0 1 2 12 2 1 7 6 4
c) 9 6 10 1 0 9 1 6 5 1 2 10
d) 4 1 8 0 1 2 5 1 5 12 2 5

18. Fie arborele cu 8 noduri și cu muchiile [2, 1], [2, 3], [4, 2], [4, 5], [6, 4], [7, 8], [4, 7]. Câți vectori de taţi distincți se pot construi pentru acest arbore? Doi vectori de taţi sunt distincți dacă există cel puțin o poziție pentru care elementele din respectivele poziții sunt distincte.

a) 8
b) 9
c) 8!
d) 72

19. Fie șirul de caractere admitere. Răsturnatul acestui șir este eretimda. Structura de date cea mai adecvată în care se poate memora un șir de caractere pentru a-l folosi răsturnat este:

a) Stiva
b) coadă
c) arbore
d) un graf orientat

20. Fie o stivă inițial vidă. Cu ajutorul subprogramelor Push(x), respectiv Pop() este adăugat elementul x, respectiv șters un element din stivă. Suma elementelor din stivă după executarea operațiilor următoare Push (3), Push (7), Push (5), Pop (), Pop (), Push (8) este:

a) 13
b) 20
c) 18
d) 11

21. Fie o coadă inițial vidă. Cu ajutorul subprogramelor Push(x) respectiv Pop() este adăugat elementul x respectiv șters un element din coadă. Care va fi suma elementelor din coadă în urma executării operațiilor următoare?
Push (3), Pop (), Push (7), Push (9), Pop (), Push (8), Push (6), Pop ()

a) 14
b) 23
c) 21
d) 17

22. Într-o coadă, inițial vidă, la fiecare pas k se introduc 2k valori şi se extrag k+1 valori. După executarea primilor 5 pași, în coadă se află un număr de elemente egal cu:

a) 15
b) 10
c) 5
d) 20

23. Se consideră un arbore cu 10 noduri, numerotate de la 1 la 10 având vectorul de tați următor (0, 1, 1, 1, 8, 8, 8, 1, 7, 7). Descendenții nodului 8 sunt:

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

24. Precizați care este lungimea maximă a unui lanț simplu (lanț în care fiecare muchie apare o singură dată) într-un arbore cu 10 noduri în care fiecare nod are gradul un număr impar.

a) 7
b) 6
c) 5

d) 4

25. Precizați care este numărul maxim de frunze ce apar într-un arbore cu 10 de noduri, dacă fiecare nod are gradul mai mic sau egal cu 3.

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

26. Precizați care este suma maximă a elementelor care apar într-un vector tată asociat unui arbore cu rădăcină format din 10 noduri, etichetate cu numere de la 1 la 10.

a) 100
b) 90
c) 81
d) 80

27. Un arbore binar este un arbore cu rădăcină în care orice nod are cel mult doi fii. Înălțimea unui arbore binar este dată de lungimea celui mai lung lanţ elementar care are una dintre extremităţi în rădăcină şi cealaltă în oricare dintre frunze. Numărul maxim de noduri dintr-un arbore binar de înălțime 5 este:

a) 31
b) 15
c) 63
d) 20

28. Precizați care este numărul maxim de frunze al unui arbore binar cu 100 noduri care are înălțimea minimă. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii).

a) 50
b) 51
c) 99
d) 69

29. Fie arborele cu rădăcină cu nodurile numerotate de la 1 la 15, reprezentat prin vectorul de taţi: (10, 8, 4, 10, 1, 4, 5, 10, 8, 0, 3, 5, 3, 12, 3). Precizaţi câte lanţuri elementare distincte de lungime 3, care pleacă din radăcină, există.

a) 3
b) 4
c) 5
d) 6

30. Precizați care dintre următoarele şiruri de numere pot reprezenta vectorul de taţi al unui arbore binar cu rădăcină. Se numeşte arbore binar un arbore în care fiecare nod are maxim doi descendenți direcți.

a) (5,0,4,0,4,5,3,1,7,1)
b) (5,8,4,0,4,5,3,6,7,8)
c) (5,8,4,0,4,9,3,6,2,7)
d) (5,8,4,0,4,5,3,2,7,8)

Tablouri

1. Cu ce trebuie completate spațiile punctate astfel încât matricea formată va fi de forma:
   1 1 1 2 3 4 3 5 7 ?

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$mat[i][j] = ...$$

   a) $i * j - j + i;$
   b) $i * j - j + 1;$
   c) $i * j - i + j;$
   d) $i * i - j * j;$

2. Cu ce trebuie completate spațiile punctate astfel încât matricea formată va fi de forma:
   3 6 9 6 9 12 9 12 15 ?

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$mat[i][j] =$.....

a) $ceil(3.14) * (i + j);$
b) $floor(3.14) * (i + j);$
c) $floor(3.14) * (i - j);$
d) $ceil(3.14) * (i - j);$

3. Cu ce trebuie completate spațiile punctate astfel încât matricea cu valori de tip int formată va fi de forma: 1 1 1 1 2 2 1 2 3 ?

$for(i = 1; i \leq 3; i ++)$

$for(j = 0; j < 3; j ++)$

$mat[i][j] =$.....

a) $sqrt(i + j + pow(i, j));$
b) $pow(i * j, sqrt(i * j));$
c) $sqrt(i * j + pow(i, j));$
d) $pow(i * j, sqrt(j * i));$

4. Cu ce trebuie completate spațiile punctate astfel încât matricea formată va fi de forma: 1 1 1 1 4 16 1 9 81 ?

$for(i = 1; i \leq 3; i ++)$

$for(j = 0; j < 3; j ++)$

$mat[i][j] =$.....

a) $pow(pow(i, j), 2);$
b) $pow(pow(j, i), 2);$
c) $pow(i * j, 2);$
d) $pow(2, i * j);$

5. Cu ce trebuie completate spațiile punctate astfel încât matricea(de tip int) formată va fi de forma: 0 0 0 0 1 1 0 1 2 ?

$for(i = 1; i \leq 3; i ++)$

$for(j = 0; j < 3; j ++)$

$mat[i][j] =$.....

a) $sqrt(i * j - j);$
b) $sqrt(i * j);$
c) $sqrt(i - j + 1);$
d) $sqrt(2 * i - j);$

6. Cum completăm spațiile punctate pentru a forma o matrice doar cu elementul 12 pe fiecare poziție?

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$mat[i][j] = ...$$

a) $floor(3.14) + floor(7.32);$
b) $ceil(3.14) + ceil(7.32);$
c) $ceil(3.14) + floor(7.32);$
d) $floor(3.14) + ceil(7.32);$

7. Care din următoarele formule nu crează aceași matrice ca celelarte?

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$mat[i][j] = ...$$

a) $ceil(3.14) + floor(7.32);$
b) $floor(3.14) + ceil(7.32);$
c) $ceil(3.14) + ceil(7.32) - 1;$
d) $floor(3.14) + floor(7.32);$

8. Ce returnează următorul algoritm știind că pentru x se citeste valoarea 1.

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$mat[i][j] = sqrt(i * j + pow(i, j));$$

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$if(x == mat[i][j])$$

$$cout \ll "x";$$

a) $xxxx$
b) xxx
c) $xxxxx$
d) Nu compilează

9. Care din urmatoarele formule **NU** creaza o matrice simetrica?

$$for(i = 1; i \leq 3; i ++)$$

$$for(j = 0; j < 3; j ++)$$

$$mat[i][j] =.....$$

a) $pow(i * 3 - j, sqrt(j + i));$
b) $pow(i * j, sqrt(j * i));$
c) $pow(i + j, sqrt(j + i));$
d) $pow(i * j, sqrt(j + i));$

10. Se citeste un număr n și o matrice M de dimensiune nxn de numere întregi de la tastatură. Să se sorteze descrescător elementele pare si crescător elementele impare de pe diagonala principală, pastrând ordinea par/impar. Toate tablourile vor fi alocate dinamic.

Exemplu:

4 3 4	5 6		6 3 4	5 6
1 5 2	1 7		1 3 2	1 7
8 9 2	2 3	devine	8 9 4	2 3
6 6 6	6 6		6 6 6	2 6
2 4 3	1 3		2 4 3	1 5

Fie programul :

$$int\ main(){$$

$$int\ n, **\ m, *\ even, *\ odd, count\_even = 0, count\_odd = 0;$$

$$cin \gg n;$$

$$m = new\ int\ *\ [n];$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$m[i] = new\ int[n];$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$for(int\ j = 0; j < n; j\ ++){$$

$$cin \gg m[i][j];$$

$$if(i == j){$$

$$if(m[i][i]\%2 == 0)$$

$$count\_even\ ++;$$

$$else.....;}}$$

$$even = new\ int[count\_even];$$

$$odd = new\ int[count\_odd];$$

$$count\_even = 0; count\_odd = 0;$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$if(m[i][i]\%2)$$

$$odd[count\_odd\ ++]\ =\ m[i][i];$$

$$else\ even[count\_even\ ++]\ =\ m[i][i];$$

$$int\ temp;$$

$$for(int\ i\ =\ 0;i\ <\ count\_odd\ -\ 1;i\ ++)$$

$$for(int\ j\ =\ i\ +\ 1;j\ <\ count\_odd;j\ ++)$$

$$if(even[i]<\ even[j]\{$$

$$temp\ =\ even[i];$$

$$even[i]=\ even[j];$$

$$even[j]=\ temp;\}$$

$$count\_even\ =\ 0;count\_odd\ =\ 0;$$

$$for(int\ i\ =\ 0;i\ <\ n;++\ i)$$

$$if(m[i][i]\%2)$$

$$m[i][i]=\ odd[cont\_odd\ ++];$$

$$else\ m[i][i]=\ even[count\_even\ ++];$$

$$for(int\ i\ =\ 0;i\ <\ n;i\ ++)\{$$

$$for(int\ j\ =\ 0;j\ <\ n;j\ ++)\{$$

$$cout\ll m[i][j]\ll"\ ";\}$$

$$cout\ll endl;\}$$

$$for(int\ i\ =\ 0;i\ <\ n;++\ i)$$

$$delete[]\quad m[i];$$

$$delete\ []m;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $count\_odd\ ++$
b) $count\_even\ ++$
c) $odd\ ++$
d) $even\ ++$

11. Se citeste un numar n de la tastatura si o matrice de intregi de dimensiune nxn.
Sa se inlocuiasca toate elementele de sub diagonala secundara cu patratul maximului de pe linia pe care se afla. Matricea va fi alocata dinamic.

Exemplu:

3 4 5 6 3 4 5 6

2 3 4 1 2 3 4 4

2 3 4 1 devine 2 3 4 4

2 4 2 4 2 4 4 4

Fie programul:

$$int\ main(){$$

$$int\ n, **\ m, *\ max\_values;$$

$$cin \gg n;$$

$$m = new\ int\ *\ [n];$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$m[i] = new\ int[n];$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$for(int\ j = 0; j < n; j\ ++){$$

$$cin \gg m[i][j];$$

$$if(j == 0\ ||\ m[i][j] > max\_values[i])$$

$$max\_values[i] = m[i][j];}$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$for(int\ j = ...; j < n; j\ ++)$$

$$m[i][j] = max\_values[i]\ *\ max\_values[i];$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$for(int\ j = 0; j < n; j\ ++){$$

$$cout \ll m[i][j] \ll " ";}$$

$$cout \ll endl;}$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$delete\ [] m[i];$$

$$delete\ []\ m;$$

$$return\ 0;}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) 0
b) i
c) n
d) $n - i$

12. Se citeste un număr n de la tastatură şi o matrice de întregi de dimensiune nxn. Să se înlocuiască toate elementele de sub diagonala secundară cu pătratul simetricului lor (față de

diagonala secundară), iar elementele de pe diagonala secundară să se sorteze descrescător. Matricea va fi alocata dinamic.

Exemplu:

3 4 5 2		3 4 5 6
2 3 4 1		2 3 4 25
2 3 4 1	devine	2 3 9 16
6 4 2 4		2 4 4 9

Fie programul:

$$int\ main(){$$

$$int\ n, **\ m, *\ max\_values;$$

$$cin \gg n;$$

$$m\ =\ new\ int\ *\ [n];$$

$$sec\ =\ new\ int[n];$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)$$

$$m[i] =\ new\ int[n];$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ n; j\ ++){$$

$$cin \gg m[i][j];$$

$$if(j\ ==\ n\ -\ i\ -\ 1)$$

$$\text{sec}\ sec\ [i]\ \ =\ m[i][j];}$$

$$int\ temp;$$

$$for(int\ i\ =\ 0; i\ <\ n\ -\ 1; i\ ++)$$

$$for(int\ j\ =\ i\ +\ 1; j\ <\ n; j\ ++)$$

$$if(\text{sec}\ sec\ [i]\ \ <\ \text{sec}\ sec\ [j]\ ){$$

$$temp\ =\ \text{sec}\ sec\ [i]\ ;$$

$$\text{sec}\ sec\ [i]\ \ =\ \text{sec}\ sec\ [j]\ ;$$

$$\text{sec}\ sec\ [j]\ \ =\ temp;}$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)$$

$$for(int\ j\ =\ n\ -\ i\ -\ 1; j\ <\ n; j\ ++)$$

$$if(j\ ==\ ...)$$

$$m[i][j] =\ \text{sec}\ sec\ [i]\ ;$$

$$else\ m[i][j] = m[n\ -\ j\ -\ 1][n\ -\ i\ -\ 1]\ *\ m[n\ -\ j\ -\ 1][n\ -\ i\ -\ 1];$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)\{$$

$$for(int\ j\ =\ 0; j\ <\ n; j\ ++)$$

$$cout\ll m[i][j]\ll\text{" ";}\}$$

$$cout\ll endl;\}$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) i
b) $n\ -\ i$
c) $n\ -\ 1$
d) $n\ -\ i\ -\ 1$

13. Se dă o matrice cu n linii şi n coloane şi elemente numere naturale. Să se determine câte elemente ale matricei se află pe linii și coloane de sumă egală.Elementul a[i,j] va fi numărat dacă suma elementelor de pe linia i este egală cu cea de pe coloana j. Datele se vor citi intr-o matrice care se va aloca dinamic

Exemplu:

5

3 1 8 5 4

7 8 5 1 2

2 2 6 7 3 devine 2

9 8 1 3 6

7 5 3 1 7

Fie programul:

$$int\ main()\{$$

$$int\ n, **\ m, *\ lin, *\ col;$$

$$cin\gg n;$$

$$m\ =\ new\ int\ *\ [n];$$

$$lin\ =\ new\ int[n];$$

$$col\ =\ new\ int[n];$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)\{$$

$$m[i] = new\ int[n];$$

$$lin[i] = 0;$$

$$col[i] = 0;\}$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$for(int\ j = 0; j < n; j\ ++)$$

$$cin \gg m[i][j];$$

$$lin[i] += m[i][j];$$

$$col[j] += m[i][j];\}$$

$$int\ nr = 0;$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$for(int\ j = 0; j < n; j\ ++)$$

$$if(\ldots)$$

$$nr\ ++;$$

$$cout \ll nr;$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $lin[i] == col[j]$
b) $lin[i] == col[i]$
c) $lin[j] == col[i]$
d) $lin[j] == col[j]$)

14. Se dă o matrice pătratică cu n linii și n coloane și elemente numere naturale mai mici decât 1000. Să se afișeze în ordine strict crescătoare valorile situate sub diagonala principală și deasupra diagonalei secundare. Dacă o valoare apare în zona respectivă de mai multe ori, se va afișa o singură dată.

Exemplu:

6

10 8 5 8 4 2

6 5 3 1 3 8

8 1 4 7 8 8 devine 1 5 6 8

5 1 9 6 6 1

8 9 10 1 3 6

8 2 3 3 9 6

Fie programul:

```
int main(){
int n,** m,* aux;
int k = 0;
cin>>n;
m = new int * [n];
aux = new int[n * n];
for(int i = 0; i < n; i ++)
m[i] = new int[n];
for(int i = 0; i < n; ++ i)
for(int j = 0; j < n; j ++)
cin>>m[i][j];
for(int i = 1; i < n - 1; i ++)
for(int j = 0; j < i &&...; j ++)
aux[k ++] = m[i][j];
for(int i = 0; i < k - 1; i ++)
for(int j = i + 1; j < k; j ++)
if(aux[i] > aux[j]){
int temp = aux[i];
aux[i] = aux[j];
aux[j] = temp;}
int l = aux[0];
cout<<l<<" ";
for(int i = 0; i < k; i ++)
if(l! = aux[i]){
cout<<aux[i]<<" ";
l = aux[i];}
```

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $j < n - i$
b) $j < n - i - 1$
c) $j < n - 1$
d) $j \leq n - i$

15. Se consideră un tablou bidimensional cu n linii şi n coloane ce conţine numere naturale cu cel mult două cifre fiecare.Să se determine ultima cifră a produsului elementelor de pe diagonala secundară cu proprietatea că sunt minime pe coloanele lor.

Exemplu:

4

3 4 90 10

25 2 7 9 devine 1

18 3 10 4

3 7 20 3

Dintre valorile de pe diagonala secundară, sunt minime pe coloanele lor 7 şi 3. Produsul lor este 21, iar ultima cifră este 1

$$int\ main()\{$$

$$int\ n, **\ m;$$

$$int\ s\ =\ 1;$$

$$cin \gg n;$$

$$m\ =\ new\ int\ *\ [n];$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)$$

$$m[i] =\ new\ int[n];$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ n; j\ ++)$$

$$cin \gg m[i][j];$$

$$for(int\ j\ =\ 0; j\ <\ n; j\ ++)\{$$

$$int\ minim\ =\ 2147483647;$$

$$int\ i;$$

$$for(i = 0; i < n; i ++)$$

$$if(m[i][j] < minim)$$

$$minim = m[i][j];$$

$$if(minim == m[\ldots][j])\{$$

$$s \mathrel{*}= minim;$$

$$s\% = 10;\}\}$$

$$cout \ll s;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $n - j$
b) $n - 1$
c) $j - 1$
d) $n - j - 1$

16. Se consideră o progresie aritmetică cu primul termen a1 și rația r, precum și un număr n. Să se afișeze o matrice pătratică cu n linii și n coloane care să conțină termenii acestei progresii astfel:
    - prima coloană va conține primii n termeni, în ordine, de sus în jos
    - a doua coloană va conține următorii n termeni, în ordine, de jos în sus
    - etc.

    Matricea va fi alocata dinamic in memorie

Fie programul:

$$int\ main()\{$$

$$int\ n, ** m, a1, r;$$

$$cin \gg a1 \gg r \gg n;$$

$$m = new\ int\ *\ [n];$$

$$for(int\ i = 0; i < n; i ++)$$

$$m[i] = new\ int[n];$$

$$for(int\ i = 0; i < n; i ++)\{$$

$$for(int\ j = 0; j < n; j ++)$$

$$if(\ldots)$$

$$m[n - i - 1][j] = a1 + (i + n * j) * r;$$

$$else\ m[i][j] = a1 + (i + n * j) * r;\}$$

$$for(int\ i = 0; i < n; i ++)\{$$

$$for(int\ j = 0; j < n; j ++)\{$$

$$cout \ll m[i][j] \ll " ";\}$$

$$cout \ll endl;\}$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $i\%2$
b) $j == i$
c) $j\%2$
d) $i\%2$

17. Se citeste un număr n de la tastatură și o matrice de întregi de dimensiune nxn. Să se înlocuiască toate elementele de deasupra diagonalei principale și secundare cu minimul din elementele care se află sub diagonala principală și secundară.

Exemplu:

5

3 1 8 5 4		3 1 1 1 4
7 8 5 1 2		7 8 1 1 2
2 2 6 7 3	devine	2 2 6 7 3
9 8 1 3 6		9 8 1 3 6
7 5 3 1 7		7 5 3 1 7

Fie programul:

$$int\ main()\{$$

$$int\ n, **\ m;$$

$$int\ s = 1;$$

$$m = new\ int\ *\ [n];$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$m[i] = new\ int[n];$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$for(int\ j = 0; j < n; j\ ++)$$

$$cin \gg m[i][j];$$

$$int\ minim = 2147483647;$$

$$for(int\ i = (n + 1)/2; i < n; i\ ++)$$

$$for(int\ j = n - i; j < i; j\ ++)$$

$$if(m[i][j] < minim)$$

$$minim = m[i][j];$$

$$for(int\ i = 0; i < ...; i\ ++)$$

$$for(int\ j = i + 1; j < n - i - 1; j\ ++)$$

$$m[i][j] = minim;$$

$$for(int\ i = 0; i < n; i\ ++)\{$$

$$for(int\ j = 0; j < n; j\ ++)\{$$

$$cout \ll m[i][j] \ll " ";\}$$

$$cout \ll endl;\}$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $(n - 1)/2$
b) $n - 1$
c) $n/2$
d) $(n - i)/2$

18. Se dă o matrice cu n linii și m coloane și un caracter c care poate fi + sau -. Să se sorteze,după linii, matricea crescător dacă semnul este + sau descrescător dacă semnul este -.Sortarea matricei după coloane este rearanjarea elementelor astfel încât, parcurgând matricea pe coloane, de la stânga la dreapta și de sus în jos,elementele sunt în ordine crescătoare/descrescătoare.

Exemplu:

5 4 +

2 4 1 3		0 4 12 21
9 8 7 6		1 6 16 29
20 19 18 16	devine	2 7 18 30
30 29 124 12		3 8 19 59
59 21 0 3		3 9 20 124

Fie programul:

$$int\ main()\{$$

$$int\ l, c, **\ m, *\ vec;$$

$$char\ sgn;$$

$$cin \gg l \gg c \gg sgn;$$

$$boll\ flag\ =\ sgn\ ==\ '+'\ ?\ true : false;$$

$$m\ =\ new\ int\ *\ [l];$$

$$vec\ =\ new\ int[c\ *\ l];$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$m[i] =\ new\ int[c];$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ c; ++\ j)$$

$$cin \gg vec[i\ *\ c\ +\ j];$$

$$for(int\ i\ =\ 0; i\ <\ l\ *\ c\ -\ 1; ++\ i)$$

$$for(int\ j\ =\ i\ +\ 1; j\ <\ ...; ++\ j)$$

$$if(vec[i] >\ vec[j] ==\ flag)\{$$

$$int\ temp\ =\ vec[i];$$

$$vec[i] =\ vec[j];$$

$$vec[j] =\ temp;\}$$

$$int\ counter\ =\ 0;$$

$$for(int\ j\ =\ 0; j\ <\ c; ++\ j)$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$m[i][j] =\ vec[counter\ ++];$$

$$for(int\ i\ =\ 0; i\ <\ l; i\ ++)\{$$

$$for(int\ j\ =\ 0; j\ <\ c; j\ ++)\{$$

$$cout \ll m[i][j] \ll " ";\}$$

$$cout \ll endl;\}$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $l\ *\ c$
b) $l\ +\ c$
c) $l\ *\ i$
d) $l\ +\ i$

19. Se dă o matrice cu n linii şi m coloane şi elemente numere naturale. Să se ordoneze liniile matricei crescător după suma elementelor.

Exemplu:

4 6

4 20 15 23 18 9		3 18 8 20 12 5
1 8 23 22 14 18	devine	1 8 23 22 14 18
17 15 13 18 12 15		4 20 15 23 18 9
3 18 8 20 12 5		17 15 13 18 12 15

Fie programul:

$$int\ main(){$$

$$int\ l, c, **\ m;$$

$$int\ *\ sums, *\ indices;$$

$$cin \gg l \gg c;$$

$$m\ =\ new\ int\ *\ [l];$$

$$sums\ =\ new\ int[l];$$

$$indices\ =\ new\ int[l];$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i){$$

$$m[i] =\ new\ int[c];$$

$$sums[i] =\ 0, indices[i] =\ i;}$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ c; ++\ j){$$

$$cin \gg m[i][j];$$

$$sums[i]\ +=\ m[i][j];}$$

$$for(int\ i\ =\ 0; i\ <\ l\ -\ 1; ++\ i)$$

$$for(int\ j\ =\ i\ +\ 1; j\ <\ l; ++\ j)$$

$$if(sums[i] >\ sums[j]){$$

$$int\ temp\ =\ sums[i];$$

$$sums[i] =\ sums[j];$$

$$sums[j] =\ sums[i];$$

$$temp\ =\ indices[i];$$

$$indices[i] =\ indices[j];$$

$$indices[j] = temp;\}$$

$$for(int\ i = 0; i < l; ++ i)\{$$

$$for(int\ j = 0; j < c; ++ j)$$

$$cout \ll m[\ldots][j] \ll " ";$$

$$cout \ll endl;\}$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) i
b) $i * indices[j]$
c) $i * indices[i]$
d) $indices[i]$

20. Se dă o matrice cu n linii şi n coloane şi elemente numere naturale. Cele două diagonale delimitează în matrice 4 zone:
    - NORD – elementele situate deasupra diagonalei principale şi deasupra celei secundare
    - EST – elementele situate deasupra diagonalei principale şi sub cea secundară
    - SUD – elementele situate sub diagonala principală şi sub cea secundară
    - VEST – elementele situate sub diagonala principală şi deasupra celei secundare

    Să se afişeze, în ordine crescătoare, sumele elementelor din cele patru zone.

Exemplu:

5

3 1 8 5 4

7 8 5 1 2

2 2 6 7 3 devine 10 18 19 20

9 8 1 3 6

7 5 3 1 7

Fie algoritmul:

$$int\ main()\{$$

$$int\ n, **\ m, *\ sums;$$

$$cin \gg n;$$

$$m = new\ int\ *\ [n];$$

$$sums = new\ int[4];$$

$$for(int\ i = 0; i < n; ++ i)\{$$

$$m[i] = new\ int[n];$$

$$sums[i] = 0;\}$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$for(int\ j = 0; j < n; ++\ j)$$

$$cin \gg m[i][j];$$

$$for(int\ i = 0; i < (n - 1)/2; i\ ++)$$

$$for(int\ j = i + 1; j < n - i - 1; j\ ++)$$

$$sums[0] += m[i][j];$$

$$for(int\ i = (n + 1)/2; i < n; i\ ++)$$

$$for(int\ j = n - i; j < i; j\ ++)$$

$$sums[1] += m[i][j];$$

$$for(int\ i = 1; i < n - 1; i\ ++)$$

$$for(int\ j = n - 1; j > i\ \&\&\ j > n - i - 1; j\ --)$$

$$sums[2] += m[i][j];$$

$$for(int\ i = 1; i < n - 1; i\ ++)$$

$$for(int\ j = 0; j < i\ \&\&j < n - i - 1; j\ ++)$$

$$sums[3] += m[i][j];$$

$$for(int\ i = 0; i < 3; ++\ i)$$

$$for(int\ j = i + 1; j < 4; ++\ j)\{$$

$$if(\ldots)\{$$

$$int\ temp = sums[i];$$

$$sums[i] = sums[j];$$

$$sums[j] = temp;\}$$

$$for(int\ i = 0; i < 4; i\ ++)$$

$$cout \ll sums[i] \ll " ";$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$delete\ []\ sums;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $sums[i] == sums[j]$
b) $sums[i] < sums[j]$
c) $sums[i] > sums[j]$

d) Nu compilează

21. Se consideră un tablou bidimensional cu n linii şi n coloane ce conţine numere întregi.Să se determine media aritmetică a elementelor strict pozitive din matrice, care sunt situate sub diagonala principală.Matricea va fi alocata dinamic

Exemplu:

4

$-\ 1\ 2\ 4\ 5$

$0\ 6\ 3\ 1$ devine 2.500

$2\ 4\ 2\ 0$

$3\ -\ 5\ 1\ -\ 3$

Fie algoritmul:

$$int\ main(\{$$

$$int\ n, **\ m;$$

$$float\ sum\ =\ 0;$$

$$int\ counter\ =\ 0;$$

$$cin \gg n;$$

$$m\ =\ new\ int\ *\ [n];$$

$$for(int\ i\ =\ 0; i\ <\ n; i\ ++)$$

$$m[i] = new\ int[n];$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$cin \gg m[i][j];$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ i; ++\ j)$$

$$if(m[i][j] > 0)$$

$$\{sum\ +=\ m[i][j]; counter\ ++\}$$

$$cout \ll ...$$

$$for(int\ i\ =\ 0; i\ <\ n; ++\ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $sum/float(counter)$;
b) $float(counter)/sum$;
c) $Nu\ compilează$
d) $sum\ +\ float(counter)$;

22. Se dă o matrice cu n linii şi m coloane şi elemente numere naturale. Să se ordoneze coloanele matricei astfel încât elementele de pe prima linie să fie ordonate crescător.

Exemplu:

4 6		
4 20 15 23 18 9		4 9 15 18 20 23
1 8 23 22 14 18	devine	1 18 23 14 8 22
17 15 13 18 12 15		17 15 13 12 15 18
3 18 8 20 12 5		3 5 8 12 18 20

Fie programul:

$$int\ main(){$$

$$int\ l, c, **\ m, *\ indices, *\ first;$$

$$cin \gg l \gg c;$$

$$m\ =\ new\ int\ *\ [l];$$

$$indices\ =\ new\ int[c];$$

$$first\ =\ new\ int[c];$$

$$for(int\ i\ =\ 0; i\ <\ l; i\ ++)$$

$$m[i] =\ new\ int[c];$$

$$for(int\ i\ =\ 0; i\ <\ c; ++\ i)$$

$$indices[i] =\ i;$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ c; ++\ j)$$

$$cin \gg m[i][j];$$

$$for(int\ i\ =\ 0; i\ <\ c; ++\ i)$$

$$first[i] =\ m[0][i];$$

$$for(int\ i\ =\ 0; i\ <\ c\ -\ 1; ++\ i)$$

$$for(int\ j\ =\ i\ +\ 1; j\ <\ c; ++\ j)$$

$$if(first[i] >\ first[j]){$$

$$int\ temp\ =\ first[i];$$

$$first[i] =\ first[j];$$

$$first[j] =\ temp;$$

$$temp\ =\ indices[i];$$

$$indices[i] = indices[j];$$

$$indices[j] = temp;\}$$

$$for(int\ i = 0; i < l; ++ i)\{$$

$$for(int\ j = 0; j < c; ++ j)$$

$$cout \ll m[i][...] \ll " ";$$

$$cout \ll endl;\}$$

$$for(int\ i = 0; i < l; ++ i)$$

$$delete\ []\ m[i];$$

$$delete\ []\ m;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $indices[i]$
b) $indices[j]$
c) j
d) Nu compilează

23. Se dă o matrice cu n linii și m coloane și elemente numere naturale și k valori naturale. Determinați pentru fiecare dintre cele k valori dacă apare pe fiecare linie a matricei.

Exemplu:

4 5		
3 7 9 9 7		DA
3 7 8 10 9		NU
8 9 5 10 7	devine	DA
3 5 4 7 9		NU
6		NU
9 4 7 8 10 7		DA

Fie programul:

$$int\ main()\{$$

$$int\ l, c, **\ m, k;$$

$$cin \gg l \gg c;$$

$$m = new\ int\ *\ [l];$$

$$for(int\ i = 0; i < l; i\ ++)$$

$$m[i] = new\ int[c];$$

$$for(int\ i = 0; i < l; ++ i)$$

$$for(int\ j = 0; j < c; ++ j)$$

$$cin \gg m[i][j];$$

$$cin \gg k;$$

$$int\ val;$$

$$for(int\ i = 0; i < k; i\ ++)\{$$

$$cin \gg val;$$

$$bool\ val\_flag = true;$$

$$for(int\ i = 0; i < l; ++\ i)\{$$

$$bool\ line\_flag = false;$$

$$for(int\ j = 0; j < c; ++\ j)$$

$$if(m[i][j] == val)$$

$$line\_flag = \ldots;$$

$$if(!\ line\_flag)$$

$$val\_flag = false;\}$$

$$if(val\_flag)$$

$$cout \ll "DA" \ll endl;$$

$$else\ cout \ll "NU" \ll endl;\}$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $true$
b) $false$
c) $Nu\ compilează$
d) val

24. Scrieţi un program care citeşte un tablou bidimensional cu m linii şi n coloane şi un număr k şi determină suma indicilor coloanelor care conţin valoarea k.

Exemplu:

3 5 4 2 4 5 − 8 1 3 7 9 4 − 2 3 10 5 4 2 37 6 7 3 13	devine	5

Coloana 2 şi coloana 3 conţin valoarea 3, iar suma lor este 5.

Fie algoritmul:

$$int\ main()\{$$

$$int\ l, c, k;$$

$$bool\ *\ cols;$$

$$cin \gg k \gg l \gg c;$$

$$cols\ =\ new\ bool[c];$$

$$for(int\ i\ =\ 0; i\ <\ c; ++\ i)$$

$$cols[i] = false;$$

$$int\ val;$$

$$for(int\ i\ =\ 0; i\ <\ l; ++\ i)$$

$$for(int\ j\ =\ 0; j\ <\ c; ++\ j)\{$$

$$cin \gg val;$$

$$if(val\ ==\ k)$$

$$cols[j] = true;\}$$

$$int\ sum\ =\ 0;$$

$$for(int\ i\ =\ 0; i\ <\ c; ++\ i)$$

$$if(cols[i])$$

$$sum\ +=\ ...$$

$$cout \ll sum;$$

$$retunr\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $1;$
b) $i;$
c) $(i\ +\ 1);$
d) $Nu\ compilează$

25. Se dă un vector x cu n elemente numere naturale, ordonate crescător, și un vector y cu m elemente, de asemenea numere naturale. Verificați pentru fiecare element al vectorului y dacă apare în x.Programul citește de la tastatură numărul n ($1 \leq n \leq 25000$), iar apoi cele n elemente ale vectorului x. Apoi se citește m ($1 \leq m \leq 200000$)și cele m elemente ale lui y.Programul va afișa pe ecran m valori 0 sau 1, separate prin exact un spațiu. A j-a valoare afișată este 1, dacă al j-lea element al șirului y apare în x, respectiv 0 în caz contrar.

Exemplu:

7

1 2 5 6 9 10 14 devine 0 1 1 1 0 0 0 1

8

8 14 9 14 16 15 4 2

Fie algoritmul:

$$int\ n, m, x[25005];$$

$$int\ main(){$$

$$cin \gg n;$$

$$for(int\ i = 0; i < n; i\ ++)$$

$$assert(cin \gg x[i]);$$

$$cin \gg m;$$

$$for(\ ; m; --\ m){$$

$$int\ y, p =- 1, st = 0, dr = n - 1;$$

$$assert(cin \gg y);$$

$$while(st <= dr){$$

$$int\ mijloc = ...;$$

$$if(x[mijloc] == y){$$

$$p = mijloc;$$

$$break;}$$

$$else$$

$$if(x[mijloc] > y)$$

$$dr = mijloc - 1;$$

$$else$$

$$st = mijloc + 1;}$$

$$if(p ==- 1)$$

$$cout \ll "0";$$

$$else\ cout \ll "1";}$$

$$cout \ll endl;$$

$$return\ 0;}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $st + dr/2$
b) $st/2$
c) $dr/2$
d) $(st + dr)/2$

26. Se citește de la tastatură un număr n și apoi n numere întregi. Să se verifice dacă elementele pare din șir sunt ordonate crescător iar cele impare descrescător. Încercați să rezolvați problema folosind cât mai puțină memorie.

Date de intrare:

- De la tastatură se citește un număr 1<=n<=10000 și n numere din intervalul [-10000, 10000]

Date de ieșire :

- Pe prima linie se va afișa "PARE DA" dacă numerele pare sunt ordonate crescător sau "PARE NU" dacă acestea nu sunt ordonate crescător.
- Pe a doua linie se va afișa "IMPARE DA" dacă numerele impare sunt ordonate descrescător sau "IMPARE NU" dacă acestea sunt ordonate crescător.

EXEMPLU:

5		$PARE\ DA$
2 7 6 10 5	devine	$IMPARE\ DA$
5		$PARE\ DA$
2 7 6 10 9	devine	$IMPARE\ NU$

Fie algoritmul:

$$int\ main(){$$

$$int\ n, a, parPrecedent =- 9998, imparPrecedent = 9999;$$

$$bool\ okPar = true, okImpar = true;$$

$$cin \gg n;$$

$$for(int\ i = 1; i \leq n; i\ ++){$$

$$cin \gg a;$$

$$if(a\%2 == 1){$$

$$if(a > imparPrecedent)$$

$$okImpar = false;$$

$$else\ imparPrecedent = a;}$$

$$else\{$$

$$if(a < parPrecedent)\ okPar = false;$$

$$else...;\}\}$$

$$if(okPar)$$

$$cout \ll "PARE\ DA\backslash n";$$

$$else\ cout \ll "PARE\ NU\backslash n";$$

$$if(okImpar)$$

$cout\ll"IMPARE\ DA\backslash n";$

$else\ cout\ll"IMPARE\ NU\backslash n";$

$return\ 0;\}$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $parPrecedent = a$
b) $imparPrecedent = a$
c) $parPrecedent = 1$
d) $imparPrecedent = 1$

27. Se dă un număr natural x ($0 < x < 1.000.000$) și două șiruri a și b, cu $n \geq 1$, respectiv $m \leq 100.000$ elemente, numere naturale, ordonate strict crescător. Să se afișeze, în ordine crescătoare, multiplii lui x care se află doar în unul dintre cele două șiruri.Fișierul de intrare date.in conține pe prima linie numărul x, pe linia a doua numărul n; urmează n numere naturale, ordonate crescător, ce pot fi dispuse pe mai multe linii. Linia următoare conține numărul m și urmează m numere naturale, ordonate crescător, ce pot fi dispuse pe mai multe linii.Fișierul de ieșire date.out va conține pe prima linie valorile determinate, separate printr-un spațiu.

Exemplu:

5

7

1 2 3 4

7 20 60 devine 5 10 60

9

3 5 7

8 9 10 12

20 24

Fie algoritmul:

$ifstream\ f("date.in");$

$ofstream\ g("date.out");$

$int\ frec[1000001], n, m, b[10][10001];$

$int\ main()\{$

$int\ x, nr = 0, p = 0, a;$

$f \gg a;$

$f \gg n;$

$$for(int\ i\ =\ 1;\ i \leq n;\ i\ ++)\{$$

$$f \gg x;$$

$$frec[x]\ ++;\}$$

$$f \gg m;$$

$$for(int\ i\ =\ 1;\ i \leq m;\ i\ ++)\{$$

$$f \gg x;$$

$$frec[x]\ ++;\}$$

$$for(int\ i\ =\ 0;\ i\ <\ 1000001;\ i\ ++)\{$$

$$if(frec[i] ==\ 1\ \&\&\ldots)\{$$

$$g \ll i \ll "\ ";$$

$$p\ ++\}\}$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $a\%i\ ==\ 0$
b) $i\%a\ ==\ 1$
c) $i\%a\ ==\ 0$
d) $Nu\ compilează$

28. Se dau mai multe numere naturale, fiecare cu cel mult 9 cifre. Să se afişeze, în ordine descrescătoare, toate cifrele care apar în numerele date.Fişierul de intrare date.in conţine cel mult 10.000 numere naturale, dispuse pe mai multe linii.Fişierul de ieşire date.out va conţine cifrele determinate, ordonate descrescător, câte 20 pe o linie, valorile de pe fiecare linie fiind separate prin spaţii. Ultima linie a fişierului poate conţine mai puţin de 20 de cifre.

Exemplu:

301941 81912 83392 776996 431446

devine 9 9 9 9 9 8 8 7 7 6 6 6 4 4 4 4 3 3 3 3 2 2 1 1 1 1 1 0

Fie programul:

$$ifstream\ f("date.in");$$

$$ofstream\ g("date.out");$$

$$int\ a[10];$$

$$int\ main()\{$$

$$int\ x;$$

$$while(f \gg x)$$

$$do\{$$

$$\ldots$$

$$x/ = 10;\}$$

$$while(x);$$

$$x = 0;$$

$$for(int\ i = 9; i >= 0; -- i)$$

$$for(int\ j = 1; j <= a[i]; ++ j)\{$$

$$g \ll i \ll " ";$$

$$x\ ++;$$

$$if(x\%20 == 0)$$

$$g \ll endl;\}$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $a[x\%10]\ ++;$
b) $a[x]\ ++;$
c) $a\ ++;$
d) $Nu\ compilează$

29. Se dau două şiruri cu câte n, respectiv m elemente. Dacă înmulțim fiecare element din primul şir cu fiecare element din al doilea şir, să se afle câte produse sunt mai mici decât p.Programul citeşte de la tastatură numerele n şi p, iar apoi n numere naturale, separate prin spații, reprezentând elementele primului şir, după care citeşte numărul m urmat de m numere naturale, reprezentând elementele celui de-al doilea şir.Programul va afişa pe ecran numărul nr, reprezentând numărul produselor mai mici decât p.

Exemplu:

5 99

1 2 3 4 5

2 devine 5

25 34

Fie programul:

$$int\ n, m, p, i, a[10001], b[10001], c[10001], nr, poz, x;$$

$$int\ main()\{$$

$$cin \gg n \gg p;$$

$$for(i = 1; i <= n; i ++)\{cin \gg x; a[x] ++;\}$$

$$cin \gg m;$$

$$for(i = 1; i <= m; i ++)\{cin \gg x; b[x] ++;\}$$

$$c[0] = b[0];$$

$$for(i = 1; i <= 9999; i ++)\ c[i] = c[i - 1] + b[i];$$

$$nr = a[0] * m;$$

$$for(i = 1; i <= 9999; i ++)\{$$

$$poz = (p - 1)/i;$$

$$if(poz < 10000)\ nr = nr + ...;$$

$$else\ nr = nr + a[i] * m;\}$$

$$cout \ll nr;$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $a[i] * c[i]$
b) $a[poz] * c[poz]$
c) $a[poz] * c[i]$
d) $a[i] * c[poz]$

30. Se dau n numere naturale. Pentru fiecare număr k dat, să se afle cea mai lungă secvenţă de numere naturale consecutive din şirul 1,2,3,...,k, astfel încât orice număr din secvenţă să nu fie prim.Fişierul de intrare date.in conține pe prima linie numărul n, iar pe a doua linie n numere naturale separate prin spații.Fişierul de ieşire date.out va conține pe linia i, primul număr din secvenţă şi lungimea secvenţei, pentru cel de-al i-lea număr de pe linia a doua a fişierului de intrare.Dacă sunt mai multe secvenţe de lungime maximă cu numere consecutive neprime, se va afişa cea cu primul număr din secvenţă minim

Exemplu:

3

4 11 30

devine

1 1
8 3
24 5

Fie programul:

$$int\ n, lmax, lung, pozc, i, j, pmax, lm[m + 1], p[m + 1];$$

$$char\ v[m + 1];$$

$$int\ main()\{$$

$$f \gg n;$$
$$v[1] = 1;$$
$$p[1] = 1;$$
$$lm[1] = 1;$$
$$v[2] = 0;$$
$$lmax = 1;$$
$$pmax = 1;$$
$$lung = 0;$$
$$for(i = 2; i <= m; i ++)$$
$$if(v[i] == 0)\{$$
$$lm[i] = lmax;$$
$$p[i] = pmax;$$
$$...;$$
$$while(j <= m)\{$$
$$v[j] = 1;$$
$$j = j + i;\}$$
$$lung = 0;\}$$
$$else\{$$
$$lung ++;$$
$$if(v[i - 1] == 0)\ pozc = i;$$
$$if(lung > lmax)\{\ lmax = lung; pmax = pozc;\}$$
$$lm[i] = lmax;$$
$$p[i] = pmax;\}$$
$$for(i = 1; i <= n; i ++)\{$$
$$f \gg k;$$
$$g \ll p[k] \ll "\ " \ll lm[k] \ll endl;\}$$
$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $j = i$
b) $j = i + 1$
c) $j ++$
d) $p[j] = i + 1$

31. Un vector cu elemente 0 sau 1 se numește alternativ dacă oricum am lua două elemente vecine în vector, cel puțin unul dintre ele este 0.Se dă un vector cu n elemente, numere naturale. Verificați dacă vectorul este alternativ.Programul citește de la tastatură numărul n, iar apoi n numere naturale, reprezentând elementele vectorului.Programul va afișa pe ecran mesajul DA, dacă vectorul este alternativ, respectiv NU în caz contrar.

Exemplu:

5

0 1 0 0 1 devine DA

Fie programul:

$$int\ main(){$$

$$int\ n, x[1005];$$

$$cin \gg n;$$

$$for(int\ i = 0; i < n; ++\ i)$$

$$cin \gg x[i];$$

$$bool\ ok = true;$$

$$for(int\ i = 0; i < n - 1; i\ ++)$$

$$if(\ldots)\ \ ok = false;$$

$$if(ok)\ \ cout \ll "DA";$$

$$else\ \ cout \ll "NU";$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $x[i + 1] * x[i + 1] == 1$
b) $x[i] * x[i + 1] == 1$
c) $x[i] * x[i] == 1$
d) $x[i + 1] * x[i] == 1$

32. Se dau două şiruri a şi b, cu n, respectiv m elemente, numere naturale, ordonate strict crescător. Să se afişeze, în ordine strict crescătoare, valorile existente în ambele şiruri.Fişierul de intrare date.in conţine pe prima linie numărul n; urmează n numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii. Linia următoare conţine numărul m şi urmează m numere naturale, ordonate strict crescător, ce pot fi dispuse pe mai multe linii.Fişierul de ieşire date.out va conţine, în ordine strict crescătoare, valorile existente în ambele şiruri. Aceste valori vor fi afişate câte 10 pe o linie, separate prin spaţii. Ultima linie poate conţine mai puţin de 10 de valori.

Exemplu:

7

1 3 4

7 20 24 60

9 devine 3 7 20 24

3 5 7

8 9 10 12

20 24

Fie programul:

$$int\ a[100005], n, b[100005], m, c[200005], k;$$

$$int\ main(){$$

$$f \gg n;$$

$$for(int\ i\ =\ 1; i\ <=\ n; ++\ i)$$

$$f \gg a[i];$$

$$f \gg m;$$

$$for(int\ i\ =\ 1; i\ <=\ m; ++\ i)$$

$$f \gg b[i];$$

$$int\ i\ =\ 1, j\ =\ 1;$$

$$k\ =\ 0;$$

$$while(i\ <=\ n\ \&\&\ j\ <=\ m)$$

$$if(a[i] < b[j])$$

$$i\ ++;$$

$$else$$

$$if(\ldots)\ \ j\ ++;$$

$$else\ \ c[++\ k] = a[i], i\ ++, j\ ++;$$

$$for(int\ i\ =\ 1; i\ <=\ k; ++\ i){$$

$$g \ll c[i] \ll "\ ";$$

$$if(i\%10\ ==\ 0)$$

$$g \ll "\ ";}$$

$$return\ 0;}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $a[i] > b[i]$
b) $a[i] > b[j]$
c) $a[j] > b[j]$
d) $a[j] > b[i]$

33. Se dau 2 mulţimi de numere naturale A și B. Să se afișeze mulţimea rezultată în urma efectuării unei operații.Programul citește de la tastatură:
    Pe prima linie 2 numere naturale N și M, reprezentând numărul elementelor mulţimii A, respectiv B, urmate de unul dintre caracterele * + - % :
    - * intersecţie A∩B.
    - + reuniune A∪B.
    - − diferența A\B.
    - % diferența simetrică (A\B)∪(B\A).

    Pe cea de-a 2-a linie N numere naturale reprezentând elementele mulţimii A.
    Pe cea de-a 3-a linie M numere naturale reprezentând elementele mulţimii B.
    Programul va afișa pe ecran elementele mulţimii rezultate, ordonate crescător, în urma efectuării operației.

Exemplu:

5 5 %

4 5 6 7 8 devine 1 2 3 6 7 8

1 2 3 4 5

Fie programul:

$$int\ a[200001], b[200001], N, M, i, j;$$

$$char\ op;$$

$$void\ QuickSort(int\ c[], int\ st, int\ dr)\{$$

$$if(st < dr)\{$$

$$int\ m = (st + dr)/2;$$

$$int\ aux = c[st];$$

$$c[st] = c[m];$$

$$c[m] = aux;$$

$$int\ i = st, j = dr, d = 0;$$

$$while(i < j)\{$$

$$if(c[i] > c[j])\{$$

$$aux = c[i];$$

$$c[i] = c[j];$$

$$c[j] = aux;$$

$$d = 1 - d;\}$$

$$i\ += d;$$

$$j\ -= 1 - d;\}$$

$$QuickSort(c, st, i - 1);$$

$$QuickSort(c, i + 1, dr);\}\}$$

```
int main(){
cin>>N>>M>>op;
for(i = 1; i <= N; i ++)
cin>>a[i];
for(i = 1; i <= M; i ++)
cin>>b[i];
QuickSort(a, 1, N);
QuickSort(b, 1, M);
switch(op){
case ' * ': {
i = 1, j = 1;
while(i <= N && j <= M){
if(a[i] == b[j])  cout<<a[i ++]<<"  ", j ++;
else if(a[i] < b[j]) i ++;
else j ++;}
break;}
case ' + ': {
i = 1, j = 1;
while(i <= N && j <= M){
if(a[i] == b[j])  cout<<a[i ++]<<"  ", j ++;
else if(a[i] < b[j]) cout<<a[i ++]<<"  ";
else cout<<b[j ++]<<"  ";}
while(i <= N){ cout<<a[i ++]<<"  ";}
while(j <= M){ cout<<b[j ++]<<"  ";}
break;}
case ' - ': {
i = 1, j = 1;
while(i <= N  && j <= M){
if(a[i] == b[j])  i ++, j ++;
else if(a[i] < b[j]) cout<<a[i ++]<<"  ";
else cout<<...}
```

$$while(i <= N)\{ cout\ll a[i\ ++]\ll"\ ";\}$$

$$break;\}$$

$$case\ '\%':\{$$

$$i = 1, j = 1;$$

$$while(i <= N \&\& j <= M)\{$$

$$if(a[i] == b[j])\ i\ ++, j\ ++;$$

$$else\ if(a[i] < b[j])\ cout\ll a[i\ ++]\ll"\ ";$$

$$else\ cout\ll b[j\ ++]\ll"\ ";\}$$

$$while(i <= N)\{cout\ll a[i\ ++]\ll"\ ";\}$$

$$while(j <= M)\{cout\ll b[j\ ++]\ll"\ ";\}$$

$$break;\}\}$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $j\ ++;$
b) $i\ ++$
c) $cout\ll b[j\ ++]$
d) $cout\ll b[j]$

34. Fie programul:

$$int\ n, a[1005];$$

$$int\ prim(int\ n)\{$$

$$if(n < 2)\ \ return\ 0;$$

$$if(n == 2)\ return\ 1;$$

$$if(n\%2 == 0)\ \ return\ 0;$$

$$for(int\ i = 3; i * i <= n; i\ += 2)$$

$$if(n\%i == 0)\ return\ 0;$$

$$return\ 1;\}$$

$$int\ main()\{$$

$$cin\gg n;$$

$$for(int\ i = 1; i <= n; ++\ i)$$

$$cin\gg a[i];$$

$$if(prim(a[p])\{$$

$$for(int\ i = p; i < n; ++\ i)$$

$$a[i] = a[i\ +\ 1];$$

$$n\ --;\}$$

$$for(int\ i = 1; i \leq n; ++\ i)$$

$$cout \ll a[i] \ll "\ ";$$

$$return\ 0;\}$$

Ce face acest program?

a) Programul afișează pe ecran, separate prin spații, elementele vectorului obținut prin ștergerea elementelor prime.
b) Programul afișează pe ecran elementele vectorului lipite unul de celălalt obținut prin ștergerea elementelor prime.
c) Programul afișează pe ecran, separate prin spații, elementele vectorului obținut prin ștergerea elementelor neprime.
d) Programul nu compilează

35. Se dau n numere întregi. Să se insereze între oricare două numere de aceeaşi paritate media lor aritmetică. Algoritmul se va relua în mod repetat până când nu se mai poate adăuga şirului niciun nou element.Programul citeşte de la tastatură numărul n, iar apoi n numere întregi, separate prin spații.Programul va afișa pe ecran pe câte o linie nouă, începând cu şirul inițial, toate șirurile distincte ce se pot construi prin metoda mai sus menționată. Fiecare şir se va scrie pe câte un rând nou.în cazul în care şirul conține două elemente consecutive egale, între acestea nu se va insera media aritmetică

Exemplu:		1 41 3 3 4 8
6	devine	1 21 41 22 3 3 4 6 8
1 41 3 3 4 8		1 11 21 31 41 22 3 3 4 5 6 7 8
		1 6 11 16 21 26 31 36 41 22 3 3 4 5 6 7 8

Fie programul:

$$int\ main()\{$$

$$int\ n, i, a[3000], j, ok;$$

$$cin \gg n;$$

$$for(i\ =\ 1; i\ <=\ n; i\ ++)$$

$$cin \gg a[i];$$

$$do\{$$

$$for(i = 1; i <= n; i ++)$$

$$cout\ll a[i]\ll" ";$$

$$cout\ll endl;$$

$$ok = 0; i = 2;$$

$$while(i <= n)$$

$$if((a[i - 1] + a[i])\%2 == 0\ and...)\{$$

$$for(j = n; j >= i; j --)$$

$$a[j + 1] = a[j];$$

$$a[i] = (a[i - 1] + a[i + 1])/2;$$

$$n ++;$$

$$i = i + 2;$$

$$ok = 1;\}$$

$$else\ i ++;$$

$$\}while(ok);$$

$$return\ 0;\}$$

Cu ce completăm spațiile punctate pentru a funcționa programul?

a) $a[i - 1] == a[i]$
b) $a[i + 1] != a[i]$
c) $a[i + 1] == a[i]$
d) $a[i - 1] != a[i]$

36. Fie programul:

$$int\ a, x, y, z, n, i, viz[1000001], maxim, nr;$$

$$int\ main()\{$$

$$a = 1;$$

$$viz[1] = 1;$$

$$for(i = 2; i <= 20; i ++)\{$$

$$a = a * 2;$$

$$viz[a - 1] = 1;\}$$

$$cin\gg n;$$

$$maxim =- 1; nr = 0;$$

$$for(i = 1; i <= n; i ++)\{$$

$$cin \gg x;$$

$$if(viz[x] == 1)\, nr ++;$$

$$else\, \{ if(nr > maxim) maxim = nr; nr = 0;\}\}$$

$$cout \ll maxim;$$

$$return\ 0;\}$$

Ce afișează programul dacă se citește elementele în această ordine: 5 4 6 7 15 88 ?

a) 1
b) 2
c) 3
d) *Nu compilează*

37. Fie programul:

$$int\ n, a[1005];$$

$$int\ main()\{$$

$$cin \gg n;$$

$$for(int\ i = 1; i <= n; ++ i)$$

$$cin \gg a[i];$$

$$int\ cnt = 0;$$

$$for(int\ i = 1; i < n; ++ i)$$

$$for(int\ j = i + 1; j <= n; j ++)$$

$$if(a[i] > a[j])\{$$

$$int\ aux = a[i];$$

$$a[i] = a[j];$$

$$a[j] = aux;\}$$

$$for(int\ i = 1; i <= n - 2; ++ i)$$

$$for(int\ j = i + 1; j <= n - 1; j ++)\{$$

$$int\ st = j + 1,\ dr = n,\ k = n;$$

$$while(st <= dr)\{$$

$$k = (st + dr)/2;$$

$$if(a[i] + a[j] > a[k])$$

$$st = k + 1;$$

$$else\ dr = k - 1;\}$$

$$cnt \mathrel{+}= dr - j;\}$$

$$cout \ll cnt \ll endl;$$

$$return\ 0;\}$$

Ce afișează programul dacă se citește elementele în această ordine: 5 3 5 10 7 6 ?

a) 7
b) 14
c) 6
d) 8

38. Fie programul:

$$int\ n, v[1005];$$

$$int\ main()\{$$

$$cin \gg n;$$

$$for(int\ i = 1; i <= n - 2; ++ i)$$

$$cin \gg v[i];$$

$$n \mathrel{-}= 2;$$

$$for(int\ i = 1; i < n; ++ i)$$

$$for(int\ j = i + 1; j <= n; ++ j)$$

$$if(v[i] > v[j])\{$$

$$int\ aux = v[i];$$

$$v[i] = v[j];$$

$$v[j] = aux;\}$$

$$int\ x =- 1, y =- 1;$$

$$if(n == 3\ \&\&\ v[2] - v[1] == v[3] - v[2]\ \&\&\ (v[2] - v[1])\%2 == 0)\{$$

$$int\ r = (v[2] - v[1])/2;$$

$$x = v[1] + r;$$

$$y = v[2] + r;\}$$

$$else$$

$$if(n == 2)\{$$

$$int\ r = (v[2] - v[1])/3;$$

$$x = v[1] + r;$$

$$y = v[1] + 2 * r;\}$$

$$else\{$$

$$int\ r\ =\ v[n]-\ v[1];$$

$$for(int\ i\ =\ 1;i\ <\ n;++\ i)$$

$$if(r\ >\ v[i\ +\ 1]-\ v[i])$$

$$r\ =\ v[i\ +\ 1]-\ v[i];$$

$$for(int\ i\ =\ 1;i\ <\ n\ \&\&\ y\ ==-\ 1;i\ ++)$$

$$if(v[i\ +\ 1]-\ v[i]!=r)\{$$

$$if(x\ ==-\ 1)\{$$

$$x\ =\ v[i]+\ r;$$

$$if(x\ +\ r\ <\ v[i\ +\ 1])$$

$$y\ =\ x\ +\ r;\}$$

$$else\ \ y\ =\ v[i]+\ r;\}\}$$

$$cout\ll x\ll"\ "\ll y\ll endl;$$

$$return\ 0;\}$$

Ce afișează programul dacă se citește elementele în această ordine: 6 13 19 7 4 ?

a) 10 15
b) 9 16
c) 10 16
d) 9 15

Siruri de caractere

1. Ce afișează programul următor ?

$$int\ main()\{$$

$$char\ s[]\ =\ "ADMITERE";$$

$$int\ k\ =\ 0,i;$$

$$for(i\ =\ 0;i\ <\ strlen(s);i\ ++)\{$$

$$if(strchr("AEIOU",s[i]))$$

$$k\ ++;\}$$

$$strcpy(s\ +\ k,"S");$$

$$cout\ll s;$$

$$return\ 0;\}$$

a) S

b) ADMITS
c) ADMMS
d) ADMIS

2. Ce afișează programul următor ?

$$int\ main(){$$

$$char\ s[]\ =\ "ADMITERE";$$

$$int\ k\ =\ 0, i;$$

$$for(i\ =\ 0; i\ <\ strlen(s); i\ ++){$$

$$if(s[i] <=\ 77)$$

$$s[i] +=\ 10;$$

$$else\ s[i] -=\ 10;}$$

$$cout \ll s;$$

$$return\ 0;}$$

a) KNCSJOHO
b) KNWSJOHO
c) eroare (programul nu compileaza)
d) ADMITERE

3. Ce afișează programul următor ?

$$int\ main(){$$

$$char\ s[]\ =\ "ADMITERE";$$

$$int\ n\ =\ strlen(s)\ -\ 1, k1\ =\ 0, k2\ =\ 0;$$

$$while(n!=\ 0){$$

$$if(strchr("AEIOU", s[n]!=\ 0)$$

$$k1\ ++;$$

$$else\ k2\ ++;$$

$$n\ --:}$$

$$s[k1]='\backslash 0';$$

$$cout \ll s[k2];$$

$$return\ 0;}$$

a) eroare(programul nu compileaza)
b) I
c) o valoare random din memorie
d) T
4. Ce afișează programul următor ?

$$int\ main(){$$

$$char\ s[10]\ =\ "ADMITERE";$$

$$cout \ll strlen(s);$$

$$s[5]='\backslash 0';$$

$$cout \ll strlen(s);$$

$$return\ 0;\}$$

a) 88
b) eroare (programul nu compilează)
c) 85
d) 89

5. Ce afișează programul următor ?

$$int\ main()\{$$

$$char\ s[15]\ =\ "ADMITERE";$$

$$cout \ll strlen(s);$$

$$s[9]='*';$$

$$s[10]='*';$$

$$cout \ll strlen(s);$$

$$return\ 0;\}$$

a) 88
b) eroare (programul nu compilează)
c) 85
d) 89

6. Ce afișează programul următor ?

$$int\ main()\{$$

$$char\ s[]\ =\ "ADMITERE";$$

$$cout \ll strlen(s);$$

$$s[9]='*';$$

$$s[10]='*';$$

$$cout \ll strlen(s);$$

$$return\ 0;\}$$

a) 88
b) eroare(programul nu compilează)
c) 85
d) 89

7. Ce afișează programul următor ?

$$int\ main()\{$$

$$char\ s[]\ =\ "ADMITERE";$$

$$cout\ll strlen(s);$$

$$s[8]='*';$$

$$cout\ll strlen(s);$$

$$return\ 0;\}$$

a) 88
b) eroare(programul nu compilează)
c) 85
d) 89

8. Care dintre următoarele secvențe de mai jos transformă caracterul salvat in variabila s din literă mică a alfabetului englez in literă mare a alfabetului englez?

a) $s\ =\ s\ +\ 32;$
b) $s\ =\ s\ -\ 32;$
c) $s='C';$
d) $s='C'\ -\ 31;$

9. Ce afișează programul următor ?

$$int\ main()\{$$

$$char\ s[]\ =\ "ADMITERE";$$

$$int\ count\ =\ 0;$$

$$for(int\ i\ =\ 0;i\ <\ strlen(s);i\ ++)\{$$

$$int\ aux\ =\ i;$$

$$while(strchr("AEIOU",s[aux])\ ==\ NULL)\{$$

$$count\ ++;$$

$$aux\ ++;\}$$

$$i\ =\ aux;\}$$

$$cout\ll count;$$

$$return\ 0;\}$$

a) Numărul subsecvențelor din șir alcătuite din cel puțin o consoană
b) Numărul subsecvențelor din șir alcătuite din cel puțin o vocală
c) Numărul consoanelor din șir
d) Numărul consoanelor unice din șir

10. Știm că variabilele a și b memorează șiruri de caractere.Pentru ce set de valori {a,b} expresia de mai jos are valoarea 1 ?

$$!!(strcmp(a,b)\ <\ 0)\&\&(a[2]\%3\ ==\ b[5]\%5\ ==\ 0);$$

a) a[] = “ADMITERE” ; b[] = “FACULTATE”
b) a[] = “TESTARE” ; b[] = “INFORMATICA”
c) a[] = “EXAMEN”; b[] = “ADMITERE”;
d) a[] = “INFORMATICA”; b[] = “FMI”;

11. Știm că variabilele a și b memorează șiruri de caractere.Pentru ce set de valori {a,b} expresia de mai jos are valoarea 1 ?

$$(strcmp(b + 5, a + 3) > 0 \&\& (abs(a[3] - b[4]) == 4);$$

a) a[] = “ADMITERE”; b[] = “UNIBUC”;
b) a[] = “EXAMEN”; b[] = “ADMITERE”;
c) a[] = “TESTARE”; b[] = “FMI”;
d) a[]=”ADMITERE”; b[] = “INFORMATICA”;

12. Considerăm adunarea șirurilor de caractere în felul următor: pentru fiecare caracter de pe aceeași poziție adunăm valorile ASCII, iar dacă ajungem la finalul alfabetului (trecem de caracterul ‘Z’), continuăm cu ‘A’ + … (considerăm literele alfabetului plasate pe un cerc)

Exemplu: $TEST + TEST = NJLN$

$A + B = C$

$DA + AAXY = EBXY$

Care va fi șirul rezultat din adunarea șirurilor “ADMITERE” si “INFORMATICA” ?

a) JRXLRLSYICA
b) JRSXLRSYADM
c) JRSXLRSYICA
d) JRSXLRSYITE

13. Fie declararea :

$$char\ s[] = "T\backslash 0EST";$$

Ce valoare are expresia

$$(strlen(s) == 5 \ \&\& \ s[2] == '\,E\,';$$

a) 1
b) 0
c) eroare (programul nu compilează)
d) o valoare random

14. Fie declararea :

$$char\ s[] = "T\backslash 0EST";$$

Ce valoare are expresia

$$(strlen(s) == 1 \ \&\& \ s[2] == '\,E\,';$$

a) 1
b) 0
c) eroare (programul nu compilează)
d) valoare random

15. Se da urmatorul program:

$$int\ main(){$$
$$char\ a[] = "ADMITERE";$$
$$char\ b[] = "INFORMATICA";$$
$$for(int\ i = 0; i < strlen(a) - 1; i\ ++)$$
$$for(int\ j = i + 1; j < strlen(a); j\ ++){$$
$$if(a[i] > a[j]{$$
$$char\ aux = a[i];$$
$$a[i] = a[j];$$
$$a[j] = aux;}}$$
$$for(int\ i = 0; i < strlen(a) - 1; i\ ++)$$
$$for(int\ j = i + 1; j < strlen(a); j\ ++){$$
$$if(b[i] > b[j]{$$
$$char\ aux = b[i];$$
$$b[i] = b[j];$$
$$b[j] = aux;}}$$
$$int\ ok = 1;$$
$$for(int\ i = 1; i < strlen(a); i\ ++)$$
$$if(a[i]! = b[i]){$$
$$ok = 0;$$
$$break;}$$
$$if(ok == 1)cout\ll"DA";$$
$$else\ cout\ll"NU";$$
$$return\ 0;}$$

Pentru ce set de valori {a,b} programul afișeaază “DA” ?

a) a[] = “TESTARE” ; b[] = “TEST”
b) a[] = “ADMITERE” ; b[] = “ADMITEERE”;
c) a[] = “TESTARE”; b[] = “ARETSET”;
d) a[] = “INFORMATICA”; b[] = “FMARCAITTACA”;

16. Ce afișează următorul program ?

$int\ main(){$

$char\ s[] = "TESTARE\_LA\_INFORMATICA";$

$for(int\ i = 0; i < strlen(s); i\ ++){$

$if(strchr("AEIOUT", s[i]))$

$strcpy(s + i + 2, s + i + 5);}$

$s[4]='\0';$

$cout \ll s;$

$return\ 0;}$

a) TEST
b) TERA
c) TESE
d) TERT

17. Ce afișează următorul program ?

$int\ main(){$

$char\ s[] = "TESTARE\_LA\_INFORMATICA";$

$char\ *\ cuv = strtok(s, "\_");$

$while(cuv != NULL){$

$cout \ll cuv \ll endl;$

$cuv = strtok(NULL, "A");}$

$return\ 0;}$

a) TESTARE
LA
INFORMATICA

b) TEST
RE
L
INFORM
TIC

c) TESTARE
L
_INFORM
TIC

d) TESTARE_
l_
INFORM_
TIC

18. Ce afișează următorul program ?

$int\ main(){$

$$char\ s[] = "FMIUNIBUC";$$

$$for(int\ i = 0; i < strlen(s); i\ ++)$$

$$s[i] = s[i] + 33$$

$$strcpy(s + 2, s + 5);$$

$$cout \ll s;$$

$$return\ 0;\}$$

a) fmibuc
b) cvd
c) jvojcvd
d) gnjcvd

19. Ce afișează următorul program ?

$$int\ main()\{$$

$$char\ s[] = "ADMITERE";$$

$$for(int\ i = 0; i < strlen(s); i\ ++)$$

$$s[i] = s[strlen(s) - 1 - i];$$

$$strcpy(s + 1, s + 5);$$

$$cout \ll s;$$

$$return\ 0;\}$$

a) EETMDA
b) EERE
c) RET
d) EMDA

20. Variabila s memorează un șir de caractere. Care dintre următoarele secvențe transformă doar al doilea ‘d’ din șir în ‘D’ ?

a) $s[2] = 'D';$
b) $for(int\ i = 0; i < strlen(s); i\ ++)\{$
   $int\ k = 0;$
   $if(s[i] == 'd')\ k\ ++;$
   $if(k == 1)\ s[i] = 'D';\}$
c) $int\ i = 0, k = 0;$
   $while(i < strlen(s))\{$
   $if(s[i] == 'd')\ k\ ++;$
   $if(k == 1) s[i] = 'D';$
   $i\ ++;\}$
d) $int\ i = 0, k = 0;$
   $while(i < strlen(s))\{$
   $if(s[i] == 'd')\ k\ ++;$
   $if(k == 2) s[i] = 'D';$

i ++;}

21. Fie următorul algoritm de criptare pe șiruri de caractere: Scriem șirul pe linii de câte K elemente, apoi reconstruim șirul concatenând elementele parcurse pe coloane.
    Exemplu : Pentru șirul ADMITEREUNIBUC și K = 3 , avem :
    ADM
    ITE
    REU
    NIB
    UC
    Și reconstruind șirul obținem: AIRNUDTEICMEUB .
    Ce valoare poate lua K astfel încât șirul criptat „EAEDAMTRXMNEDIEE" să aibă sens? (să fie format din cuvinte din limba română, separate prin spațiu)
    a) K = 4
    b) K = 3
    c) K = 2
    d) K = 1

22. Ce afiseaa programul?

$$int\ main(){$$

$$char\ x[30] = "dimineata", y[30] = "min";$$

$$strcat(y, x + strlen(x) - 1);$$

$$cout\ll y;$$

$$char * p = strstr(y, x);$$

$$if(p! = NULL)$$

$$cout\ll(p - y);$$

$$else\ cout\ll"0";}$$

    a) 0
    b) ta
    c) 3
    d) Dieta

23. Ce afiseaza urmatorul program?

$$int\ main(){$$

$$char\ s[256] = "UNIBUC", t[256];$$

$$int\ x = 3;$$

$$strcpy(t, s + x + 1);$$

$$strcpy(s + x, t);$$

$$cout\ll s;}$$

a) UC
b) BUC
c) UNIBUC
d) UNIUC

24. Ce afiseaza urmatorul program?

$$int\ main(){$$

$$char\ s[256] = "UNIBUC", t[256];$$

$$int\ x = 3;$$

$$strcpy(t, s + x);$$

$$strcpy(s + x + 1, t);$$

$$s[x]='A';$$

$$cout\ll s;\}$$

a) UC
b) UNIABUC
c) BUC
d) UABUC

25. Ce afiseaza programul?

$$int\ main(){$$

$$int\ i = 0;$$

$$char\ s[11] = "abaemeiut", t[11];$$

$$cout\ll strlen(s);$$

$$while(i < strlen(s))$$

$$if(strchr("aeiou", s[i]! = NULL){$$

$$strcpy(t, s + i + 1);$$

$$strcpy(s + i, t);$$

$$i\ ++;\}$$

$$else\ i = i + 2;$$

$$cout\ll" "\ll s;\}$$

a) 9 $bemeut$
b) 9 $emeut$
c) 9 $bemeu$
d) 9 $emeu$

1. Un algoritm de tip backtracking generează în ordine lexicografică toate permutările mulțimii {1, 2, 3, 4, 5}. După generarea permutarii 2-4-5-3-1 care va fi urmatoarea permutare generata?

a) $2 - 5 - 1 - 3 - 5$
b) $2 - 4 - 3 - 5 - 1$
c) $2 - 1 - 3 - 4 - 5$
d) $2 - 1 - 3 - 5 - 4$

2. Un algoritm de tip backtracking generează în ordine lexicografică toate permutările mulțimii {a, b, c, d, e}. Care este a 5-a solutie generata de acest algoritm?

a) $a - b - e - d - c$
b) $a - b - d - e - c$
c) $a - c - d - b - e$
d) $a - b - e - c - d$

3. Utilizând metoda backtracking se generează toate numerele cu câte 3 cifre impare, cifre care aparţin mulţimii {7, 8, 1, 6, 2, 3}. Primele 3 soluţii generate sunt, in aceasta ordine: 777, 771, 773, 717. Cea de-a 7-a soluţie generată este:

a) 731
b) 737
c) 137
d) 173

4. Utilizând metoda backtracking se generează, în ordine crescătoare, toate numerele naturale pare cu trei cifre, cu proprietatea că nu există două cifre egale alăturate şi suma cifrelor este 10. Primele 4 numere generate sunt, în această ordine: 136, 154, 172,190. Al 8-lea număr generat este:

a) 460
b) 262
c) 316
d) 406

5. Utilizând metoda backtracking se generează, în ordine crescătoare, toate numerele naturale impare cu trei cifre, cu proprietatea că nu există două cifre egale alăturate şi suma cifrelor este 12. Primele 4 numere generate sunt, în această ordine: 129, 147, 165, 183.
Al 10-lea număr generat este:

a) 291
b) 273
c) 309
d) 327

6. Care din urmatoarele afirmatii legate de metoda Backtracking sunt adevarate:
1. este o metoda lenta
2. este o metoda costisitoare
3. este o metoda de complexitate mare
4. este o metoda rapida
5. solutia se construieste element cu element
6. verificarea conditiei de continuare garanteaza obtinerea unei solutii rezultat

a) 1, 2, 3, 5, 7
b) 2, 3, 4, 5, 6
c) 1, 2, 3, 5, 6

d) 1, 2, 3, 7

7. Fie multimea de litere [a, b, c, d]. Se genereaza permutarile acestei multimi. Precizati care sunt solutiile anterioara si urmatoare solutiei cabd:

a) bdac si cbad
b) bdca si cadb
c) bdca si cdba
d) bcda si cdba

8. Un algoritm de tip backtracking genereaza, in ordine, toate permutarile unei multimi cu 5 elemente. Primele 4 solutii generate sunt: 1 2 3 4 5, 1 2 3 5 4, 1 2 4 3 5, 1 2 4 5 3. Care este a 5-a solutie generata din acest algoritm?

a. 1 3 2 4 5
b. 1 3 2 5 4
c. 1 3 4 2 5
d. 1 2 5 3 4

9. 19. Fie utilizarea metodei Backtracking pentru generarea tuturor numerelor palindrom formate din 4 cifre din Mulțimea {1,2,3}. Numărul soluțiilor:

a) 6
b) 12
c) 3
d) 9

10. Avem n persoane la o coada pentru a cumpara bilete, persoana 0 e primul la coada, persoana n-1 este in spatele liniei. Avem n=4 si un vector t=[5,1,1,1] unde t[i] reprezinta numarul de bilete pe care trebuie sa cumpere persoana i. Fiecare persoana poate sa cumpere doar un bilet cand este primul la coada, dupa se intoarce la sfarsitul cozii si asteapta sa cumpere urmatorul bilet. Fiecarei persoane ii ia un minut sa cumpere un bilet. Daca o persoana si-a cumparat toate biletele necesare, iese din coada. Cate minute ii ia fiecarei persoane incepand de la pozitia 0 sa cumpere toate biletele necesare.

a) [8, 2, 3, 4]
b) [8, 2, 3, 5]
c) [7, 2, 3, 4]
d) [8, 2, 4, 4]

11. Stiva utilizată în implementarea parcurgerii în lățime are rolul:

a) să rețină varfurile care au fost descoperite și urmează să fie prelucrate;
b) nu se folosește stiva la parcurgerea în lățime;
c) să rețină ordinea de vizitare a varfurilor;
d) să rețină varfurile care au fost deja vizitate pentru a evita vizitarea repetată;

12. Care din urmatoarele afirmatii **NU** corespunde metodei Greedy (metoda optimului local):

a) problema poate fi imaginata ca o multime A cu n elemente
b) pot exista mai multe submultimi diferite acceptabile (solutii posibile), dintre care una este
considerata solutie optima pe baza unui criteriu care trebuie maximizat (minimizat)
c) o solutie posibila este o submultime (B) care indeplineste o conditie data (B este acceptabila)
d) problema se descompune in probleme de complexitate mai mica sau probleme cu rezolvare imediata

13. Care din urmatoarele afirmatii legate de sortarea crescatoare prin interclasarea unei secvente de numere reale este adevarata:
a) insereaza un element intr-un vector ordonat pe pozitia corecta
b) este denumita si sortarea prin interschimbare
c) utilizeaza metoda Divide Et Impera
d) determina minimul din vector si il insereaza pe pozitia corecta

14. Care din urmatoarele afirmatii referitoare la metoda Divide et Impera sunt adevarate?
1. este utilizata in rezolvarea unor probleme complexe
2. implementarea este realizata de obicei prin subprograme recursive
3. se aplica pentru problemele care pot fi descompuse in probleme cu complexitate mai mica
4. rezolvarea problemelor rezultate in urma descompunerii este mai usoara decat rezolvarea
intregii probleme initiale
5. pentru fiecare din problemele rezultate in urma descompunerii se aplica un procedeu diferit de descompunere

a) 1,2,3,4,5
b) 1,3,4,5
c) 2,3,4,5
d) 1,2,3,4

15. Care din urmatoarele afirmatii **NU** este adevarata:
a) la recursivitatea directa, apelul recursiv se realizeaza prin intermediul mai multor functii care se apeleaza circular
b) un subprogram recursiv genereaza (cel putin) un apel catre el insusi
c) un algoritm iterativ sau recursiv poate fi implementat printr-un subprogram iterativ sau recursiv
d) recursivitatea directa poate fi simpla sau multipla

16. Ce afiseaza urmatorul program?

$$int\ main()$$

$$\{int\ x\ =\ 7;$$

$$cout \ll x\ ++;$$

$cout \ll ++ x;\}$

a) 79
b) 80
c) 81
d) 78

Barem

Capitolul 1:Sintaxa

1	2	3	4	5	6	7	8	9	10	11	12
a	d	a	b	d	b	c	d	d	a	c	a

Capitolul 2:Grafuri

1	2	3	4	5	6	7	8	9	10	11	12
b	d	c	d	c	b	b	b	c	b	d	b

13	14	15	16	17	18	19	20	21	22	23	24
b	b	d	c	c	a	d	a	c	c	b	a

25	26	27	28	29	30	31	32	33	34	35	36	37
b	b	c	c	d	d	a	c	d	d	b	d	b

Capitolul 3:Subprogram

1	2	3	4	5	6	7	8	9	10	11
a	b	a	c	a	d	a	c	d	a	b

12	13	14	15	16	17	18	19	20	21	22
c	a	a	a	b	b	b	b	d	c	a

Capitolul 4:Structuri de date

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c	b	a	d	b	d	c	c	c	c	d	a	a	c	b

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
a	c	b	a	d	a	b	d	c	a	b	d	a	c	b

Capitolul 5:Tablouri

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	b	c	a	a	b	d	c	a	a	d	d	a	b	d	c	a	a	d

20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
c	a	b	a	c	d	a	c	a	d	b	b	b	a	a	d	b	a	c

Capitolul 6:Siruri de caractere

1	2	3	4	5	6	7	8	9	10	11	12
d	b	d	c	a	b	d	b	c	a	d	c

13	14	15	16	17	18	19	20	21	22	23	24	25
b	a	c	b	c	d	b	d	c	a	d	b	a

Capitolul 7: Algoritmi si metode de programare

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	a	b	c	d	c	b	d	d	a	d	d	c	d	a	a