

Universitatea „Dunărea de Jos” din Galați

CULEGERE DE TESTE PENTRU ADMITEREA 2014

DISCIPLINA: INFORMATICĂ

CULEGEREA DE TESTE ESTE RECOMANDATĂ PENTRU CANDIDAȚII CARE VOR SUSȚINE CONCURS DE ADMITERE LA DOMENIILE/SPECIALIZĂRILE URMĂTOARELOR FACULTĂȚI:

- Mecanică
- Arhitectură navală
- Automatică, Calculatoare, Inginerie Electrică și Electronică
- Ingineria Materialor și a Mediului
- Inginerie din Brăila
- Economie și Administrarea Afacerilor

1. O funcție reprezintă o secvență de cod care:
 - A. poate fi apelată doar din funcția principală *main()*;
 - B. poate fi apelată doar dintr-un fișier extern fișierului sursă;
 - C. poate fi apelată din diferite părți ale unei aplicații.

2. Instrucțiunile unui program editat în limbajul C:
 - A. trebuie să aparțină funcției *main()*;
 - B. pot fi editate în orice zonă a programului;
 - C. trebuie să aparțină unei funcții.

3. Execuția fiecărui program în limbajul C începe cu execuția instrucțiunilor din:
 - A. funcția *main()*;
 - B. prima funcție editată;
 - C. funcția care are declarate variabile globale.

4. Rolul compilatorului este acela de a testa dacă:
 - A. un program este editat corect;
 - B. dacă execuția unui program conduce la un rezultat corect;
 - C. un program este corect editat și conduce la rezultatul dorit de utilizator.

5. Care dintre următorii identificatori sunt editați corect pentru limbajul C:
 - A. `-x`;
 - B. `corect#1`;
 - C. `unu_si_doi`.

6. În limbajul C, următoarea construcție: `#include<specificator_fișier>` permite includerea unui fișier:
 - A. definit de utilizator, care trebuie să se găsească în același director cu fișierul sursă dacă nu este prevăzută "calea" sau într-un director ce corespunde "căii" specificate;
 - B. definit de utilizator, care însă nu trebuie să se găsească în același director cu fișierul sursă, chiar și atunci când nu este prevăzută "calea";
 - C. standard.

7. În limbajul C, următoarea construcție: `#include "specificator_fișier"` permite includerea unui fișier:
 - A. definit de utilizator, care trebuie să se găsească în același director cu fișierul sursă dacă nu este prevăzută "calea", sau într-un director ce corespunde "căii" specificate;
 - B. definit de utilizator, care însă nu trebuie să se găsească în același director cu fișierul sursă, chiar și atunci când nu este prevăzută "calea";
 - C. definit de utilizator, care însă trebuie să se găsească în același director cu fișierul sursă.

8. Secvența de cod alăturată, editată în limbajul C este greșită deoarece:

- A. funcția *main()* nu poate fi de tipul void ;
- B. corpul oricărei funcții trebuie inclus între acolade;
- C. funcția *main()* nu are niciun parametru, deci nu trebuia inclus *void* între paranteze.

```
void main(void)
printf("%d",100);
```

9. În limbajul C, datele variabile:

- A. își schimbă valorile în timpul execuției programului;
- B. nu își schimbă valorile în timpul execuției programului;
- C. își schimbă valorile doar la compilare.

10. În limbajul C, inițializarea se referă la:

- A. atribuirea unei valori unei variabile pe parcursul execuției programului;
- B. atribuirea unei valori unei variabile în momentul în care aceasta este declarată;
- C. atribuirea unei valori unei variabile la prima instrucțiune de atribuire.

11. În limbajul C, funcția *getch()* permite citirea:

- A. cu ecou (la citire, caracterul este afișat) a unui caracter de la tastatură;
- B. fără ecou a unui caracter de la tastatură ;
- C. cu ecou a unui șir de caractere de la tastatură.

12. În limbajul C, funcția *getche()* permite citirea:

- A. cu ecou (la citire, caracterul este afișat) a unui caracter de la tastatură ;
- B. fără ecou a unui caracter de la tastatură ;
- C. cu ecou a unui șir de caractere de la tastatură.

13. Funcția *scanf()* permite:

- A. citirea datelor de la tastatură sub controlul unor formate specifice;
- B. citirea datelor de la tastatură, fără includerea unor formate specifice;
- C. afișarea datelor sub controlul unor formate specifice.

14. În limbajul C, ++ este un operator de:

- A. incrementare, deci poate crește sau scădea valoarea unui variabile cu 1;
- B. decrementare, deci poate crește sau scădea valoarea unui variabile cu 1;
- C. incrementare, poate sa apăre atât după numele unei variabile, cât și înaintea lui și produce același rezultat: crește cu 1 valoarea variabilei incrementate.

15. În limbajul C, -- este un operator de:

- A. incrementare, deci poate crește sau scădea valoarea unui variabile cu 1;
- B. decrementare, deci poate crește sau scădea valoarea unui variabile cu 1;
- C. decrementare, poate apărea atât după numele unei variabile, cât și înaintea lui și produce același rezultat: scade cu 1 valoarea variabilei decrementate.

16. Fie următoarea expresie: $a=x++$. Care dintre următoarele afirmații este adevărată:

- A. incrementează x cu 1, după care valoarea incrementată este atribuită variabilei a ;
- B. atribuie variabilei a valoarea variabilei x , după care se incrementează x cu 1;
- C. nu există o regulă în acest sens.

17. Fie următoarea expresie: $a=++x$. Care dintre următoarele afirmații este adevărată:
- A. inițial se realizează incrementarea cu 1 a valorii variabilei x , după care valoarea incrementată este atribuită variabilei a ;
 - B. inițial, se atribuie variabilei a valoarea variabilei x , după care se incrementează cu 1 valoarea variabilei x ;
 - C. nu există o regulă în acest sens.
18. Instrucțiunea *for* este o instrucțiune repetitivă:
- A. cu contor, ceea ce înseamnă că variabila care controlează ciclul este testată doar la sfârșitul iterației;
 - B. cu contor, ceea ce înseamnă că variabila care controlează ciclul este testată doar la începutul iterației;
 - C. cu contor, ceea ce înseamnă că variabila care controlează ciclul este testată la fiecare iterație.
19. *While* este o instrucțiune repetitivă:
- A. cu test inițial;
 - B. cu test final;
 - C. cu contor.
20. *Do while* este o instrucțiune repetitivă:
- A. cu test inițial;
 - B. cu test final;
 - C. cu contor.
21. Diferența dintre instrucțiunile repetitive *while* și *do while* este aceea că:
- A. *while* este una cu contor în timp ce *do while* este cu test final;
 - B. *while* este una cu test inițial în timp ce *do while* este cu contor;
 - C. *while* este una cu test inițial în timp ce *do while* este cu test final.
22. În limbajul C, în cazul unei structuri alternative de tipul *switch*, variabila testată poate fi:
- A. doar de tipul *int*;
 - B. de tipul *int* sau *char*;
 - C. de orice tip numeric.
23. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?
- A. eroare la compilare;
 - B. eroare la execuție;
 - C. $a=10; b=20$.
- ```
int a=10,b=20;
printf("a=%d, b=%d",a++,b++);
```
24. Care este rezultatul ce va fi afișat după executarea secvențe de cod în limbajul C?
- A.  $a=10, b=20$ ;
  - B.  $a=11, b=21$ ;
  - C. eroarea la compilare.
- ```
int a=10,b=20;
printf("a=%d, b=%d",++a,++b);
```

25. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=10, b=20;
- B. a=9, b=19;
- C. eroarea la compilare.

```
int a=10,b=20;
printf("a=%d, b=%d",--a,--b);
```

26. Care este rezultatul afișat după execuția secvenței alăturate de cod în limbajul C?

- A. a=10, b=20;
- B. a=9, b=19;
- C. eroarea la compilare.

```
int a=10,b=20;
printf("a=%d, b=%d",a--,b--);
```

27. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=10, b=20, a=10, b=20;
- B. a=10, b=20, a=11, b=21;
- C. a=11, b=21, a=10, b=20.

```
int a=10,b=20;
printf("a=%d, b=%d, a=%d, b=%d",a++,b++,a,b);
```

28. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=10, b=20, a=10, b=20;
- B. a=10, b=20, a=9, b=19;
- C. a=9, b=19, a=10, b=20.

```
int a=10,b=20;
printf("a=%d, b=%d, a=%d, b=%d",a--,b--,a,b);
```

29. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=10, b=20, a=10, b=20;
- B. a=10, b=20, a=11, b=21;
- C. a=11, b=21, a=10, b=20.

```
int a=10,b=20;
printf("a=%d, b=%d, a=%d, b=%d",++a,++b,a,b);
```

30. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=10, b=20, a=10, b=20;
- B. a=10, b=20, a=9, b=19;
- C. a=9, b=19, a=10, b=20.

```
int a=10,b=20;
printf("a=%d, b=%d, a=%d, b=%d",--a,--b,a,b);
```

31. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=11, b=19, x=11, y=19;
- B. a=11, b=19, x=10, y=20;
- C. a=10, b=20, x=10, y=20.

```
int a=10,b=20,x,y;
x=a++;
y=b--;
printf("a=%d, b=%d, x=%d, y=%d",a,b,x,y);
```

32. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. a=11, b=19, x=11, y=19;
- B. a=11, b=19, x=10, y=20;
- C. a=10, b=20, x=10, y=20.

```
int a=10,b=20,x,y;  
x=++a;  
y=-b;  
printf("a=%d, b=%d, x=%d, y=%d",a,b,x,y);
```

33. În limbajul C, aplicarea operatorului % permite obținerea:

- A. câtului împărțirii a două numere întregi;
- B. restului împărțirii a două numere întregi;
- C. adresei la care este memorată o variabilă.

34. În limbajul C, operatorul & se aplică:

- A. oricărui tip de dată și furnizează adresa la care este memorată o variabilă;
- B. datelor de tip întreg și furnizează adresa la care este memorată o variabilă;
- C. doar datelor de tip întreg și returnează restul împărțirii a două numere întregi.

35. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. x=2;
- B. x=1;
- C. x=2,08.

```
int a = 25 , b = 12 , x;  
x = a % b ;  
printf("x=%d",x);
```

36. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 0 1 2 3 4;
- B. 0 1 2 3 4 5;
- C. 0.

```
int n=5,i;  
for(i=0;i<n;i++)  
printf("%d ",i);
```

37. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 1 2 3 4 5 6 7 8 9 10;
- B. 1 2 3 4 5 6 7 8 9;
- C. 2 4 6 8 .

```
int n=10,i;  
for(i=1;i<n;i++)  
if(i%2==0)  
printf("%d ",i);
```

38. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 1 2 3 4 5 6 7 8 9 10;
- B. 1 2 3 4 5 6 7 8 9;
- C. 1 3 5 7 9.

```
int n=10,i;  
for(i=1;i<n;i++)  
if(i%2!=0)  
printf("%d ",i);
```

39. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 20;
- B. 45;
- C. 25.

```
int n=10,i,s=0;  
for(i=1;i<n;i++)  
if(i%2==0)  
s+=i;  
printf("%d ",s);
```

40. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 20;
- B. 45;
- C. 25.

```
int n=10,i,s=0;
for(i=1;i<n;i++)
    if(i%2!=0)
        s+=i;
printf("%d ",s);
```

41. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 1 2 3 4 5 6 7 8 9 10;
- B. 10 9 8 7 6 5 4 3 2 1 0;
- C. 10 9 8 7 6 5 4 3 2 1.

```
int i,n=10;
for(i=n;i>0;i--)
    printf("%d ",i);
```

42. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. eroare la compilare deoarece lipsește secvența de inițializare a variabilei de control din instrucțiunea *for*;
- B. 45;
- C. 25.

```
int i=1,s=0,n=10;
for(;i<n;i++)
    s+=i;
printf("%d",s);
```

43. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. eroare la compilare deoarece lipsește secvența de inițializare a variabilei de control în cadrul instrucțiunii *for*;
- B. eroare la compilare, în instrucțiunea *for* nu poate lipsi secvența care incrementează valoarea variabilei de control;
- C. 10.

```
int i=1,s=0,n=5;
for(;i<n;)
    s+=i++;
printf("%d",s);
```

44. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. eroare la compilare deoarece în cadrul instrucțiunii *for* nu există nicio expresie; este obligatoriu ca în sintaxa acestei instrucțiuni să se regăsească cel puțin o expresie;
- B. eroare la compilare deoarece din cadrul instrucțiunii *for* nu poate lipsi secvența care incrementează variabila de control;
- C. 10.

```
int i=1,s=0,n=5;
for(;;){
    if(i>=n)
        break;
    s+=i++;
}
printf("%d",s);
```

45. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 30;
- B. 160;
- C. 190.

```
int i,x=1,y=20;
int z=5,s=0;
for(i=x;i<y;i++){
    if(i%z==0)
        s+=i;
}
printf("%d",s);
```

46. Care este rezultatul ce va fi afișat după secvenței alăturate de cod în limbajul C?

- A. 1;
- B. 2;
- C. 1 2 4 5 10.

```
int i,y=20;
for(i=1;i<y;i++)
    if(y%i==0)
        printf("%d ",i);
```

47. Care este rezultatul ce va fi afișat după secvenței alăturate de cod în limbajul C?

- A. 3 4 6 7 8 9;
- B. 1 3 4 6 7 8 9;
- C. 2 5.

```
int i,y=10;
for(i=1;i<=y;i++)
    if(y%i!=0)
        printf("%d ",i);
```

48. Care este rezultatul ce va fi afișat după secvenței alăturate de cod în limbajul C?

- A. 7;
- B. 37;
- C. 8.

```
int i,y=10,s=0;
for(i=1;i<y;i++)
    if(y%i==0) s+=i;
printf("%d",s);
```

49. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 6;
- B. 36;
- C. 16.

```
int i,y=20,s=0;
for(i=1;i<y;i++)
    if(y%i==0&& i%2==0)
        s+=i;
printf("%d",s);
```

50. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 16;
- B. 6;
- C. 8.

```
int i,y=20,s=0;
for(i=1;i<=y;i++)
    if(y%i==0&& i%2!=0)
        s+=i;
printf("%d",s);
```

51. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 6 15;
- B. 7 11 13;
- C. 7 9 11 13 15.

```
int i,j,x=6,y=15,c;
for(i=x;i<=y;i++){
    c=0;
    for(j=2;j<i;j++)
        if(i%j==0) c=1;
    if(c==0)
        printf("%d ",i);
}
```

52. Care este rezultatul ce va fi afișat după secvenței alăturate de cod în limbajul C?

- A. 0;
- B. 31;
- C. 40.

```
int i,j,x=6,y=15,c,s=0;
for(i=x;i<=y;i++){
    c=0;
    for(j=2;j<i;j++)
        if(i%j==0)
            c=1;
    if(c==0) s+=i;
}
printf("%d ",s);
```


53. Care este rezultatul ce va fi afișat după secvenței alăturate de cod în limbajul C?

- A. 7 9 11 13 15;
- B. 6 8 9 10 12 14;
- C. 6 8 9 10 12 14 15.

```
int i,j,x=6,y=15,c,s=0;
for(i=x;i<=y;i++){
    c=0;
    for(j=2;j<i;j++)
        if(i%j==0)
            c=1;
    if(c==1)
        printf("%d ",i);
}
```

54. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 28;
- B. 42;
- C. 70.

```
int i,x=5,y=30,z=7,s=0;
for(i=x;i<=y;i++){
    if(i%z==0&& i%2!=0)
        s+=i;
}
printf("%d",s);
```

55. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 28;
- B. 42;
- C. 70.

```
int i,x=5,y=30,z=7,s=0;
for(i=x;i<=y;i++){
    if(i%z==0&& i%2==0)
        s+=i;
}
printf("%d",s);
```

56. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 23;
- B. 33;
- C. 8.

```
int i,n=10,s=0;
for(i=1;i<n;i++){
    if(i%3==0||i%5==0)
        s+=i;
}
printf("%d",s);
```

57. Care este rezultatul ce va fi afișat după executarea secvenței de cod în limbajul C?

- A. 2 4 5 7 8 10;
- B. 2 4 5 7 8;
- C. 1 2 4 5 7 8 10.

```
int i=1,n=10;
do {
    i++;
    if(i%3!=0)
        printf("%d ",i);
}while(i<n);
```

58. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3 6 9;
- B. 2 4 5 7 8;
- C. 1 2 4 5 7 8.

```
int i=1,n=10;
while(i<n){
    if(i%3!=0)
        printf("%d ",i);
    i++;
}
```

59. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3;
- B. 9;
- C. 18.

```
int i=0,n=11,s=0;
while(i<n){
    i++;
    if(i%3==0) s+=i;
}
printf("%d",s);
```

60. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3;
- B. 9;
- C. 18.

```
int i=0,n=12,s=0;
do {
    if(i%3==0) s+=i;
    i++;
}while(i<n);
printf("%d",s);
```

61. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3;
- B. 9;
- C. 30.

```
int i=0,n=12,s=0;
do {
    i++;
    if(i%3==0)
        s+=i;
}while(i<n);
printf("%d",s);
```

62. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 10;
- B. 15;
- C. 55.

```
int i=0,n=10,s=0;
do {
    i++;
    s+=i;
}while(s<n);
printf("%d",s);
```

63. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 10;
- B. 15;
- C. 55.

```
int i=0,n=10,s=0;
while(s<=n){
    i++;
    s+=i;
};
printf("%d",s);
```

64. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 0;
- B. 14;
- C. 21.

```
int i=0,s=0,c=7;
do {
    i++;
    s+=i;
}while(s%c!=0);
printf("%d",s);
```

65. Dacă utilizatorul introduce întregii: 2, 3, 4, 5, 6, care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 10;
- B. 20;
- C. 12.

```
int i,j,nr,s=0,c;
for(i=1;i<=5;i++){
    c=0;
    printf("numarul %d: ",i);
    scanf("%d",&nr);
    for(j=2;j<nr;j++)
        if(nr%j==0)
            c=1;
    if(c==0)
        s=s+nr;
}
printf("%d",s);
```

66. Dacă utilizatorul introduce întregii: 5, 6, 7, 8, 9, care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 35;
- B. 21;
- C. 23.

```
int i,j,nr,s=0,c;
for(i=1;i<=5;i++) {
    c=0;
    printf("numarul %d: ",i);
    scanf("%d",&nr);
    for(j=2;j<nr;j++)
        if(nr%j==0)
            c=1;
    if(c==1) s=s+nr;
}
printf("%d",s);
```

67. Ce realizează secvența alăturată de cod sursă editată în limbajul C?

- A. introducerea de numere întregi până când se vor introduce trei numere impare;
- B. introducerea de numere întregi până când se vor introduce trei numere prime;
- C. introducerea a trei numere întregi.

```
int i=0,j,nr,s=0,c,contor=0;
do {
    c=0;
    i++;
    printf("numarul %d: ",i);
    scanf("%d",&nr);
    for(j=2;j<nr;j++)
        if(nr%j==0)c=1;
    if(c==0)
        contor++;
} while(contor<3);
```

68. Ce realizează secvența alăturată de cod sursă în C?

- A. calculează produsul numerelor a și b;
- B. calculează a^b ;
- C. calculează b^a .

```
int a=3,b=4,i,p=a;
for(i=1;i<b;i++)
    p=p*a;
```

69. Dacă utilizatorul introduce cinci numere întregi, ce face secvența alăturată de cod sursă editată în limbajul C?

- A. calculează maximul numerelor introduse;
- B. calculează minimul numerelor introduse;
- C. verifică dacă s-a introdus un întreg cu valoarea 1.

```
int nr,i,m;
for(i=1;i<=5;i++) {
    printf("nr%d=",i);
    scanf("%d",&nr);
    if(i==1) m=nr;
    else
        if(m<nr) m=nr;
}
```

70. Dacă utilizatorul introduce întregii: 3, 11, 5, 9, 7, care este valoarea finală a variabilei m ce va fi afișată?

- A. $m=3$;
- B. $m=11$;
- C. $m=9$.

```
int nr,i,m;
for(i=1;i<=5;i++){
    printf("nr%d=",i);
    scanf("%d",&nr);
    if(i==1) m=nr;
    else
        if(m<nr) m=nr;
}
printf("m=%d",m);
```

71. Dacă utilizatorul introduce întregii: 3, 11, 5, 9, 7, care este valoarea finală a variabilei m ce va fi afișată?

- A. $m=3$;
- B. $m=11$;
- C. $m=9$.

```
int nr,i,m;
for(i=1;i<=5;i++){
    printf("nr%d=",i);
    scanf("%d",&nr);
    if(i==1) m=nr;
    else
        if(m>nr) m=nr;
}
printf("m=%d",m);
```

72. Dacă utilizatorul introduce cinci numere întregi, ce realizează secvența alăturată de cod sursă C?

- A. calculează maximul numerelor introduse;
- B. calculează minimul numerelor introduse;
- C. verifică dacă s-a introdus un întreg cu valoarea 1.

```
int nr,i,m;
for(i=1;i<=5;i++){
    printf("nr%d=",i);
    scanf("%d",&nr);
    if(i==1) m=nr;
    else
        if(m>nr) m=nr;
}
```

73. Pentru următoarea secvență, precizați varianta în care se găsesc întregi ce nu permit finalizarea structurii repetitive:

- A. 15, 30, 45;
- B. 3, 6, 9, 12;
- C. 3, 6, 8, 10.

```
int nr;
do {
    scanf("%d",&nr);
}while(nr%3!=0&&nr%5!=0);
```

74. În limbajul C, domeniul de vizibilitate al unei variabile globale este:
- doar funcția *main()*;
 - prima funcție declarată, sau funcția *main()* dacă nu există funcții definite de utilizator;
 - atât funcția *main()* cât și orice altă funcție definită de utilizator.
75. În limbajul C, dacă se declară două variabile cu același nume, una globală și alta locală, atunci în cadrul funcției unde este declarată variabila locală:
- compilatorul va lua în considerare valoarea variabilei locale;
 - compilatorul va lua în considerare valoarea variabilei globale;
 - se va produce o eroare la compilare.

76. Ce rezultat va produce secvența alăturată de cod C:

- `x=10;`
- `x=100;`
- eroare la compilare deoarece s-au declarat două variabile cu același nume.

```
void f(){
    int x=100;
    printf("x=%d",x);
}
int x=10;
main(){
    f();
}
```

77. Ce rezultat va produce secvența alăturată de cod C:

- `x=1;`
- `x=10;`
- `x=11.`

```
int x=10;
void f(int a){
    x=1;
    printf("x=%d",x);
}
main(){
    f(x++);
}
```

78. Ce rezultat va produce secvența alăturată de cod C:

- `x=10 a=10;`
- `x=11 a=11;`
- `x=1 a=10.`

```
int x=10;
void f(int a)
{
    x=1;
    printf("x=%d a=%d",x,a);
}
main(){
    f(x++);
}
```

79. Ce rezultat va produce secvența alăturată de cod C:

- `x=10 a=10;`
- `x=10 a=11;`
- `x=11 a=10.`

```
int x=10;
void f(int a)
{
    printf("x=%d a=%d",x,a);
}
main(){
    f(x++);
}
```

80. Ce rezultat va produce secvența alăturată de cod C:

- A. x=21 a=21;
- B. x=21 a=11;
- C. x=21 a=10.

```
int x=10;
void f(int a){
    x+=a;
    printf("x=%d a=%d",x,a);
}
main(){
    f(x++);
}
```

81. Ce rezultat va produce secvența alăturată de cod C:

- A. x=11 a=10;
- B. x=10 a=1;
- C. x=1 a=10.

```
int x=10;
void f(int a){
    x = x - a;
    printf("x=%d a=%d",x,a);
}

main(){
    f(x++);
}
```

82. Ce rezultat va produce secvența alăturată de cod C:

- A. a=10;
- B. a=20;
- C. a=30.

```
void f(int a,int b){
    a=a+b;
    printf("a=%d",a);
}
main(){
    f(10,20);
}
```

83. Ce rezultat va produce secvența alăturată de cod C:

- A. eroare la compilare deoarece se apelează funcția cu parametrii actuali x și y , în timp ce în funcția apelată parametrii formali sunt a și b ;
- B. a=5;
- C. a=15.

```
void f(int a,int b){
    a=a+b;
    printf("a=%d",a);
}
main(){
    int x=5, y=10;
    f(x,y);
}
```

84. Ce rezultat va produce secvența alăturată de cod C:

- A. eroare la compilare deoarece se apelează funcția cu parametrii actuali x și y , în timp ce în funcția apelată parametrii formali sunt a și b ;
- B. eroare la compilare deoarece la apel se folosește un nume de variabilă și o valoare întreagă;
- C. a=15.

```
void f(int a,int b){
    a=a+b;
    printf("a=%d",a);
}
main(){
    int x=5;
    f(x,10);
}
```

85. Care sunt parametrii formali ai funcției *f*:

- A. a și b;
- B. x și y;
- C. a, b, x, y.

```
void f(int a,int b){
    int x,y;
}
main(){
    f(10,20);
}
```

86. Care sunt parametrii actuali ai funcției *f*:

- A. a și b;
- B. x și y;
- C. 10, 20.

```
void f(int a,int b){
    int x,y;
}
main(){
    f(10,20);
}
```

87. Care sunt tipurile de date permise pentru parametrii formali ai unei funcții?

- A. doar tipuri întregi;
- B. doar tipul *int*;
- C. orice tip de dată valid în C.

88. Care va fi rezultatul afișat pe ecran după executarea secvenței alăturate de cod C:

- A. y=20;
- B. y=10;
- C. nu se va afișa nimic.

```
int f(int a,int b){
    int x=a,y=b;
    return x;
    printf("y=%d",y);
}
main(){
    int a;
    a=f(10,20);
}
```

89. Care va fi rezultatul afișat pe ecran după executarea secvenței alăturate de cod C:

- A. a=10;
- B. a=100;
- C. a=30.

```
int f(int a,int b){
    int x=a,y=b;
    return x+y;
}
main(){
    int a=100;
    a=f(10,20);
    printf("a=%d",a);
}
```

90. Dacă utilizatorul introduce a=2 și b=5, care este rezultatul ce se va obține după rularea secvenței alăturate de cod C:

- A. x=7;
- B. x=14;
- C. x=9.

```
int f(int a,int b){
    int s=0,i;
    for(i=a;i<=b;i++)
        s+=i;
    return s;
}
main(){
    int a,b,x;
    printf("a="); scanf("%d",&a);
    printf("b="); scanf("%d",&b);
    if(a>b) x=f(b,a);
    else
        if(b>a) x=f(a,b);
        else x=0;
    printf("x=%d",x);
}
```

91. Dacă utilizatorul introduce a=5 și b=2, care este rezultatul ce se va obține după rularea secvenței alăturate de cod C:

- A. x=7;
- B. x=14;
- C. x=9.

```
int f(int a,int b){
    int s=0,i;
    for(i=a;i<=b;i++)
        s+=i;
    return s;
}
main(){
    int a,b,x;
    printf("a="); scanf("%d",&a);
    printf("b="); scanf("%d",&b);
    if(a>b) x=f(b,a);
    else
        if(b>a) x=f(a,b);
        else x=0;
    printf("x=%d",x);
}
```

92. Dacă utilizatorul introduce a=5 și b=5, care este rezultatul ce se va obține după rularea secvenței alăturate de cod C:

- A. x=7;
- B. x=14;
- C. x=0.

```
int f(int a,int b) {
    int s=0,i;
    for(i=a;i<=b;i++)
        s+=i;
    return s;
}
main() {
    int a,b,x;
    printf("a="); scanf("%d",&a);
    printf("b="); scanf("%d",&b);
    if(a>b) x=f(b,a);
    else
        if(b>a) x=f(a,b);
        else x=0;
}
```



```
printf("x=%d",x);  
}
```

93. Dacă utilizatorul introduce doar numere întregi în funcția principală *main()* pentru variabila *a*, ce face programul alăturat?

- A. apelează funcția până când utilizatorul introduce o valoare pentru variabila *a* care să fie mai mare sau egală decât 100;
- B. apelează la infinit funcția *f()*;
- C. calculează suma tuturor întregilor introduși de utilizator și afișează ultima valoare a lui *S* care este mai mică decât 100.

```
int f(int nr){  
    static int S=0;  
    S=S+nr;  
    return S;  
}  
main(){  
    int a,S;  
    do {  
        printf("a="); scanf("%d",&a);  
        S=f(a);  
    }while(S<100);  
    printf("S=%d",S-a);  
}
```

94. Dacă utilizatorul introduce în funcția principală *main()* următoarele numere întregi pentru variabila *a*: 10, 35, 45, 9, 1, care este valoarea variabilei *S* afișată la final?

- A. S=10;
- B. S=100;
- C. S=99.

```
int f(int nr) {  
    static int S=0;  
    S=S+nr;  
    return S;  
}  
main() {  
    int a,S;  
    do {  
        printf("a="); scanf("%d",&a);  
        S=f(a);  
    }while(S<100);  
    printf("S=%d",S-a);  
}
```

95. Dacă utilizatorul introduce în funcția principală *main()* următoarele numere întregi pentru variabila *a*: 10, 35, 45, 9, 1, care este valoarea variabilei *S* afișată la final?

- A. S=10;
- B. S=100;
- C. S=99.

```
int f(int nr){  
    static int S=0;  
    S=S+nr;  
    return S;  
}  
main() {  
    int a,S;  
    do {  
        printf("a=");scanf("%d",&a);  
        S=f(a);  
    }while(S<100);  
    printf("S=%d",S);  
}
```

96. Dacă utilizatorul introduce în funcția principală *main()* următoarele numere întregi pentru variabila *a*: 10, 35, 45, 10, 1, care este valoarea variabilei *S* afișată la final?

- A. S=10;
- B. S=100 și permite introducerea ultimului întreg (a numărului 1);
- C. S=100 și nu mai permite introducerea ultimului întreg (a numărului 1).

```
int f(int nr) {
    static int S=0;
    S=S+nr;
    return S;
}
main() {
    int a,S;
    do {
        printf("a=");scanf("%d",&a);
        S=f(a);
    }while(S<100);
    printf("S=%d",S);
}
```

97. Dacă utilizatorul introduce două numere întregi, care este rezultatul programului alăturat?

- A. dacă $a=b$ afișează $a=b$, dacă $a>b$ afișează $a>b$, iar dacă $a<b$ afișează $a<b$;
- B. verifică doar dacă $a=b$; $a=b$ egale, afișează $a=b$, iar dacă nu sunt egale, nu afișează nimic;
- C. eroare la execuție.

```
main() {
    int a,b;
    printf("a="); scanf("%d",&a);
    printf("b="); scanf("%d",&b);
    if(a=b) printf("a=b");
    else
        if(a>b) printf("a>b");
        else printf("b<a");
}
```

98. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3;
- B. 4;
- C. nu va afișa nimic.

```
int a=3;
if(++a<4)
    if(++a<4)
        printf("%d",a);
    else
        printf("%d",a);
```

99. Care este rezultatul ce va fi afișat după executarea secvenței alăturate de cod în limbajul C?

- A. 3;
- B. 4;
- C. 5.

```
int a=3;
if(++a<4){
    if(++a<4) printf("%d",a);
}
else printf("%d",a);
```

100. Dacă se introduce un număr întreg, ce rezultat are execuția programului alăturat?

- A. verifică dacă numărul introdus este par sau impar, afișează mesajul corespunzător și valoarea lui *a* introdusă;
- B. verifică numai dacă numărul introdus este par, afișează mesajul corespunzător după care afișează valoarea lui *a* introdusă; dacă s-a introdus un număr impar, nu afișează nimic;

```
main() {
    int a;
    printf("a="); scanf("%d",&a);
    if(a%2==0)
        printf("Numar par!")
        printf("%d",a);
    else
        printf("Numar impar!")
        printf("%d",a);
}
```

C. eroare la compilare.

101. Care va fi valoarea variabilei x după efectuarea secvenței alăturate de cod:

- A. 8;
- B. 9;
- C. 10.

```
int x,y;
x=3+17/3;
y=x+1;
x=y+1;
```

102. Specificați care va fi conținutul tabloului a după execuția secvenței alăturate:

- A. 2 2 2 B. 2 3 4 C. 2 3 4
- 4 4 4 2 3 4 3 4 5
- 6 6 6 2 3 4 4 5 6

```
int a[3][3];
for(i=0;i<3;i++)
for(j=0;j<3;j++)
a[i][j]=i+j+2;
```

103. Considerăm un tablou cu n linii și m coloane în care toate elementele primei linii sunt nule. Știind că nu există alt element egal cu zero (nesituat pe prima linie), câte elemente nenule sunt în tablou?

- A. (m+n)-n;
- B. (m+n)-m;
- C. n*m-m.

104. Care este valoarea minimă pe care o poate avea variabila x astfel încât instrucțiunea cu test final din secvența alăturată să efectueze o singură iterație ?

- A. 18;
- B. 20;
- C. 19.

```
int x,y;
x=?
y=10;
execută
scrie y;
y=y-1;
cât timp x>2*y;
stop.
```

105. Care dintre instrucțiuni are ca efect citirea unui caracter din fișierul "c.txt" ?

- A. f=fopen("c.txt","r"); fclose(f);
- B. f=fopen("c.txt","r");scanf("%c",&x); fclose(f);
- C. f=fopen("c.txt","r");scanf(f,"%c",&x); fclose(f).

106. Care este valoarea minimă pe care o poate avea variabila x astfel încât instrucțiunea *while* să nu efectueze nici o iterație?

- A. 3;
- B. 4;
- C. 6.

```
int x,y;
x=?; y=10;
while (2*x-1<y)
{ scrie y;
x=x+1;}
stop.
```

107. Ce se va afișa la executarea secvenței alăturate?

- A. 01234;
- B. 12340;
- C. 1234.

```
a=1;
while(a%5!=0){
cout<<a%5;
a++;
}
```

108. Se consideră declarațiile alăturate. Care dintre următoarele referiri ale elementelor celor 2 vectori sunt incorecte?

- A. $a[i*2]$;
- B. $a[-i]$;
- C. b .

```
float a[600];
char b[300];
unsigned char i;
```

109. Considerăm un tablou cu n linii și m coloane. Câte elemente sunt situate pe marginea tabloului (prima și ultima linie, prima și ultima coloană)?

- A. $2*m+2*n$;
- B. $m*n-2*n-2*m$;
- C. $2*n+2*m-4$.

110. Ce se va afișa în urma execuției secvenței alăturate de instrucțiuni ?

- A. 121 11;
- B. 120 12;
- C. 120 11.

```
a=10; b=a+1;
if(a!=b)b++;
else a++;
a=b*a;
cout<<a<<" "<<b;
```

111. Ce valori vor fi afișate în urma rulării programului alăturat?

- A. 10 20 30 40 40;
- B. 0 10 20 30 40;
- C. 10 10 20 20 30.

```
int a[5],i;
for(i=0;i<5;i++)
    a[i]=(i+1)*10;
for(i=1;i<5;i++)
    a[i]-=a[i-1];
for(i=0;i<5;i++)
    cout<<a[i]<<" ";
```

112. Care dintre următoarele variante realizează deschiderea la citire a fișierului text "A.txt"?

- A. $f=fopen("A.txt","w");$
- B. $f=fopen("r","A.txt");$
- C. $f=fopen("A.txt","r");$

113. Se consideră declarațiile alăturate. Care dintre următoarele referiri ale elementelor celor 2 vectori sunt incorecte?

- A. $x[y[1]]$;
- B. $y((x[1]))$;
- C. $x[1]$.

```
char x[301];
unsigned char y[301], i;
```

114. Se consideră secvența alăturată. Ce se va afișa în urma executării instrucțiunilor secvenței?

- A. 1;
- B. 3;
- C. 6.

```
int a,s;
s=0; a=0;
do {
    a=a+1;
    s=s+a;
} while (s>=10);
cout<<s;
```

115. Ce valori vor fi afișate în urma rulării programului alăturat?

- A. 0 1 0 2 0 3;
- B. 1 0 2 0 3 0;
- C. 0 2 0 4 0 6.

```
int a[6],i;
for(i=0;i<6;i++)a[i]=0;
for(i=0;i<3;i++)
a[2*i+1]=i+1;
for(i=0;i<6;i++)
cout<<a[i]<<" ";
```

116. Ce valoare are expresia: $E=a/b/c*d-a$ unde $a=36$, $b=6$, $c=3$, $d=4$?

- A. 36.0;
- B. 40.0;
- C. -28.0.

117. Ce se va afișa la execuția secvenței alăturate?

- A. 2 4 8 6;
- B. 2486;
- C. 24816.

```
p=1;
do{
p*=2;
cout<<p;
}while(p%10!=6);
```

118. Fie declarația `int v[4]`. Specificați valoarea elementelor tabloului după execuția instrucțiunii *for*:

- A. 2 4 4 8 10;
- B. 0 2 4 6 8;
- C. 0 1 2 3 4.

```
for(i=0;i<5;i++) v[i]=2*i;
```

119. Se consideră declarația `unsigned char a[4]`; Specificați valoarea elementelor tabloului după execuția instrucțiunii *for*:

- A. 1 1 1 1 1;
- B. 0 0 0 0 0;
- C. 0 1 2 3 4.

```
for(i=0;i<5;i++) a[i]=i;
```

120. Determinați valoarea expresiei alăturate:

- A. 14;
- B. 2;
- C. 0.

```
(int)fabs(-14.2))%7
```

121. Presupunem că variabila întregă a are valoarea 10, iar variabila întregă b are valoarea 5. Ce se va afișa în urma executării secvenței alăturate de instrucțiuni?

- A. 5 5;
- B. 10 5;
- C. 5 1.

```
aux=1;
if(a<b)
aux=a;
a=b;
b=aux;
cout<<a<<" "<<b;
```

122. Care dintre următoarele variante realizează deschiderea la scriere a fișierului text "B.txt"?

- A. `f=fopen("B.txt","r");`
- B. `f=fopen("B.txt","w");`
- C. `f=fopen("w","B.txt");`

123. Se consideră următoarea secvență de program. Ce se va afișa în urma execuției instrucțiunilor secvenței?

- A. *;
- B. **;
- C. ***.

```
int a,b;
a=10;b=2;
while (a<15) {
    if (a%b==0)
    cout<< "*";
    a=a+1;
    b=b+1;
}
```

124. Știind că variabila x este utilizată pentru a memora media aritmetică a două numere naturale cu cel mult 4 cifre, stabiliți care este declarația corectă a variabilei x?

- A. `unsigned int x;`
- B. `float x[2][4];`
- C. `double x.`

125. Considerați următoarele declarații. Care dintre referirile de mai jos nu reprezintă un caracter?

- A. `x.b;`
- B. `y.b[1];`
- C. `y.a.`

```
typedef struct {
    float a;
    char b, c[10];
} Exemplu;
Exemplu x,y;
```

126. Fie declarația alăturată făcută pentru a memora numele și vârsta elevilor dintr-o clasă, în ordine alfabetică. Specificați care dintre referirile de mai jos reprezintă inițiala numelui fiecărui elev din catalog?

- A. `a[0].elev.num;`
- B. `a[0].elev.num[0];`
- C. `a[0].num[0].`

```
typedef struct {
    char nume[256];
    int varsta;
} elev;
typedef elev clasa[35];
clasa a;
```

127. Indicați care dintre următoarele antete de funcții este corect sintactic.

- A. `int F1(int z);`
- B. `int F2(int x,y);`
- C. `float F4(int x, int y).`

128. Care dintre enunțurile de mai jos sunt adevărate?

- A. algoritmul este o succesiune infinită de pași;
- B. algoritmul este o succesiune finită de operații care conduc la rezolvarea unei probleme;
- C. algoritmul este o noțiune ce nu poate fi descrisă.

129. Care este forma convențională de reprezentare a algoritmului?
A. scheme electrice;
B. scheme orare;
C. scheme logice.
130. Ce formă geometrică are blocul pentru introducerea datelor?
A. pătrat;
B. dreptunghi;
C. trapez.
131. Care din proprietățile algoritmului semnifică rezolvarea unei clase de probleme?
A. discret;
B. finit;
C. universal.
132. Ce se înscrie în blocul de decizie?
A. datele inițiale;
B. condiția;
C. rezultatul.
133. Ce trebuie să posede executantul pentru a îndeplini un algoritm?
A. putere;
B. set de numere;
C. set de instrucțiuni.
134. Ce reprezintă dreptunghiul în schema logică ?
A. calcul;
B. decizie;
C. afișarea datelor.
135. Care descriere corespunde proprietății discrete a algoritmului?
A. descrie cu precizie ordinea operațiilor care se vor efectua;
B. procesul rezolvării să fie divizat într-o succesiune de pași;
C. algoritmul să fie valabil pentru o clasă de probleme.
136. Blocul de decizie este reprezentat prin:
A. pătrat;
B. trapez;
C. romb.
137. Care din următoarele secvențe reprezintă un algoritm corect de adunare a două numere?
A. se citesc valorile a două numere; se calculează suma; se afișează rezultatul;
B. se calculează suma; se afișează rezultatul; se citesc valorile a două numere;
C. se afișează rezultatul; se calculează suma; se citesc valorile a două numere.

138. Care este tipul de dată în care se poate memora valoarea 0x100350 ?

- A. char;
- B. int;
- C. long int.

139. Care este tipul de dată în care se poate memora valoarea 100350. ?

- A. char;
- B. int;
- C. float.

140. Care dintre următoarele tipuri de date se reprezintă pe 32 de biți:

- A. char;
- B. int;
- C. long int.

141. Care este efectul secvenței alăturate de program:

- A. afișează 2/4/..98/;
- B. afișează 2/4/..100/;
- C. afișează 2/4/..98/.

```
int i;
for (i=1;i<100;++i)
    printf(i%7?"":"%d/ ",i);
```

142. Care este efectul secvenței alăturate de program:

- A. afișează 4/ 8/..128/;
- B. afișează 2/ 4/ 8/..124/;
- C. afișează 2/ 4/ 8/..128/.

```
int i=1,*j; j=&i;
do printf("%d/ ",(*j)*=2);
while(*j<100);
```

143. Care este efectul secvenței alăturate de program:

- A. afișează 98/96/..2/;
- B. afișează 100/96/..0/;
- C. afișează 98/96/..0/.

```
int i=100;
for (i--;i>0;i--)
    printf("%d/ ",--i);
```

144. Care este efectul secvenței alăturate de program:

- A. afișează 2/4/..100/;
- B. afișează 1/3/5/..100/;
- C. afișează 1/3/5/..98/.

```
int i;
for (i=1;i<100;i++)
    printf("%d/ ",++i);
```

145. Care este efectul secvenței alăturate de program:

- A. afișează 3/8/13/..93/;
- B. afișează -2/6/14/..94/;
- C. afișează -2/3/8/13/..93/.

```
int i;
for(i=1;i<100;i+=8)
    printf("%d/ ",i-=3);
```


146. Câți biți sunt alocați pentru variabila următoare: *long int unu*;
A. 8;
B. 16;
C. 32.
147. Câți biți sunt alocați pentru variabila următoare: *char doua*;
A. 8;
B. 16;
C. 32.
148. Câți biți sunt alocați pentru variabila următoare: *double sapte*;
A. 16;
B. 32;
C. 64.
149. Câți biți sunt alocați pentru variabila următoare: *short int două*;
A. 8;
B. 16;
C. 32.
150. Câți biți sunt alocați pentru variabila următoare: *float trei*;
A. 8;
B. 16;
C. 32.
151. Câți biți sunt alocați pentru variabila următoare: *long double patru*;
A. 16;
B. 32;
C. 80.
152. Câți biți sunt alocați pentru variabila următoare: *signed long int cinci*;
A. 8;
B. 16;
C. 32.

153. Care este dimensiunea alocată în memorie variabilei declarate alăturat:

- A. 12b;
B. 12B;
C. 11B.

```
char ocupa[ ]="abracadabra";
```

154. Care este dimensiunea alocată în memorie variabilei declarate alăturat:

- A. 540B;
B. 1080B;
C. 2160B.

```
float ocupa[45][12];
```

155. Care este dimensiunea alocată în memorie variabilei declarate alăturat:

- A. 7650b;
- B. 30600b;
- C. 61200b.

```
char ocupa[45][17][10];
```

156. Funcția apelată poate transmite valori funcției apelante:

- A. prin variabile static intern definite în funcția apelată;
- B. prin parametrii formali ai funcției apelate;
- C. prin valoarea returnată de funcția apelată.

157. Care din următoarele instrucțiuni sunt greșite?

- A. float total=suma=valoare=0;
- B. char litera='65';
- C. double raza=12.345E-1.

158. Ce simbol termină o linie C?

- A. .;
- B. ,;
- C. ;.

159. Ce caracter nu este valid în C?

- A. \$;
- B. !;
- C. ^.

160. Care din următoarele variante este un scallar în C?

- A. union;
- B. function;
- C. double.

161. Ce nume de variabilă este invalid?

- A. integer;
- B. Xx;
- C. net-total.

162. O instrucțiune C/C++ a cărei executare determină memorarea în variabila reală x a mediei aritmetice a valorilor variabilelor reale x, y și z este:

- A. $x=(x + y + z)/2$;
- B. $x=x/1/3 + y/1/3 + z/1/3$;
- C. $x=x/3+(y + z)/2$.

163. Dacă n este un numar natural iar u, c1, c2, s1, și s2 sunt variabile de tip întreg, ce afișează secvența alăturată?

- A. media cifrelor numărului n;
- B. suma cifrelor pare și a celor impare ale lui n;
- C. media cifrelor pare și media cifrelor impare ale

```
c1=0; c2=0; s1=0; s2=0;
while(n!=0) {
    u=n%10; n=n/10;
    if(u%2==0){c1++; s1=s1+u;}
    else {c2++;s2=s2+u;}
}
```

numărului n.

```
cout<<(float)s1/c1  
cout<<(float)s2/c2
```

164. Dacă n este un număr natural iar c și i sunt variabile de tip întreg, ce afișează secvența alăturată?

- A. cifrele numărului n și numărul acestora;
- B. divizorii lui n și numărul acestora;
- C. cifrele număr prim ale lui n și numărul acestora.

```
c=0;  
for(int i=2;i<=n/2;i++)  
    if(n%i==0){  
        cout<<i; c++;  
    }  
cout<<c;
```

165. Care este rezultatul afișat de instrucțiunile alăturate?

- A. 153;
- B. 163;
- C. 203.

```
float s=0, p=1;  
for(int i=1;i<=5;i++) {  
    p=p*i; s=s+p;  
}  
cout<<s<<endl;
```

166. Știind că a este un vector având n elemente numere naturale iar i, j și c sunt variabile de tip întreg, identificați prelucrarea realizată de secvența alăturată.

- A. afișarea elementelor care sunt numere pare;
- B. afișarea elementelor care sunt numere impare;
- C. afișarea elementelor care sunt numere prime.

```
for(i=1;i<=n;i++) {  
    c=1;  
    if(a[i]<2) c=0;  
    j=2;  
    while(j<=a[i]/2 && c==1)  
        if(a[i]%j==0) c=0;  
    else j++;  
    cout<<a[i]<<" ";  
}
```

167. Știind că a este un vector având n elemente numere întregi iar i, u și x sunt variabile de tip întreg, ce afișează secvența alăturată?

- A. suma elementelor vectorului cifră cu cifră;
- B. suma cifrelor fiecărui element al vectorului;
- C. suma cifrelor tuturor elementelor vectorului.

```
for(i=1;i<=n;i++){  
    x=a[i]; s=0;  
    while(x!=0){  
        u=x%10;  
        x=x/10;  
        s=s+u;  
    }  
    cout<<s<<"\n";  
}
```

168. Știind că a este un vector având n elemente numere întregi iar i, c și max sunt variabile de tip întreg, ce afișează secvența alăturată?

- A. elementele maxime ale vectorului;
- B. numărul de elemente egale cu maximul;
- C. suma elementelor vectorului egale cu maximul.

```
max=a[1];  
for(i=1;i<=n;i++)  
    if(a[i]>max)  
        x=a[i];  
c=0;  
for(i=1;i<=n;i++)  
    if(a[i]==max) c++;  
cout<<c;
```

169. Știind că a este un vector având n elemente numere întregi iar i, s și t sunt variabile de tip întreg, ce afișează secvența alăturată?

- A. suma elementelor pare și impare ale vectorului;
- B. sumea elementelor pare și suma elementelor impare ale vectorului;
- C. suma elementelor de pe poziții pare și suma elementelor de pe poziții impare.

```
for(i=1;i<=n;i++)
  if(i%2==1)
    s=s+a[i];
  else t=t+a[i];
cout<<"s="<<s;
cout<<"t="<<t;
```

170. Știind că a este un vector având n elemente numere întregi iar i și aux sunt variabile de tip întreg, ce prelucrare realizează secvența alăturată?

- A. permutarea circulară cu n poziții a elementelor vectorului;
- B. permutarea circulară dreapta cu o poziție a elementelor vectorului;
- C. permutarea circulară stânga cu o poziție a elementelor vectorului.

```
for(i=0;i<n;i++)
  a[i]=(i+1)*10;
aux=a[0];
for(i=0;i<n-1;i++)
  a[i]=a[i+1];
a[n-1]=aux;
```

171. Ce valori vor fi afișate în urma executării secvenței alăturate?

- A. 10 20 40 50 0;
- B. 40 10 20 30 0;
- C. 20 30 40 50 0.

```
int a[5],i,k;
n=5; k=2;
for(i=0;i<n;i++)
  a[i]=(i+1)*10;
for(i=k;i<n-1;i++)
  a[i]=a[i+1];
a[n-1]=0;
for(i=0;i<n;i++)
  cout<<a[i]<<' ';
```

172. Fie secvența de instrucțiuni alăturată. Știind că a este un vector având n elemente numere întregi, identificați prelucrarea realizată de aceasta.

- A. înlocuirea cu 0 a primelor k elemente ale vectorului a;
- B. înlocuirea cu 0 a ultimelor k elemente ale vectorului a;
- C. înlocuirea cu 0 a ultimelor i elemente ale vectorului a.

```
int i,k;
cin>>k;
if(k>0 || k<n)
  for(i=n-k;i<n;i++)
    a[i]=0;
```

173. Fie secvența de instrucțiuni alăturată. Ce valori vor fi afișate în urma executării acestei secvențe de instrucțiuni?

- A. 0 1 2 3;
- B. 0 2 4 6;
- C. 1 1 3 3.

```
int a[4][4], i, j;
for(i=0;i<4;i++) {
  for(j=0;j<4;j++)
    a[i][j]=i+j;
  min=a[i][0];
  for(j=0;j<4;j++)
    if(min>a[i][j])
      min=a[i][j];
}
```

```
cout<<min<<' ';  
}
```

174. Fie secvența de instrucțiuni alăturată. Ce valori vor fi afișate în urma executării acestei secvențe de instrucțiuni?
- A. 0 1 2 3 4 0 1 2 3;
 - B. 0 1 2 1 2 3 2 3 4;
 - C. 1 2 3 4 1 2 3 4 5.

```
int a[3][3], i, j, k;  
for(i=0;i<3;i++)  
  for(j=0;j<3;j++)  
    a[i][j]=i+j;  
k=0;  
for(j=0;j<3;j++)  
  for(i=0;i<3;i++) {  
    b[++k]=a[i][j];  
    cout<<b[k-1]<<' ';  
  }
```

175. Fie secvența de instrucțiuni alăturată. Ce valori vor fi afișate în urma executării acestei secvențe de instrucțiuni?
- A. 0 1 2 3;
 - B. 3 4 5 6;
 - C. 3 3 4 4.

```
int a[4][4], i, j;  
for(j=0;j<4;j++) {  
  for(i=0;i<4;i++) a[i][j]=i+j;  
  max=a[0][j];  
  for(i=0;i<4;i++)  
    if(max<a[i][j])  
      max=a[i][j];  
  cout<<max<<' ';  
}
```

176. Fie secvența de instrucțiuni alăturată. Ce valori vor fi afișate în urma executării acestei secvențe de instrucțiuni?
- A. 1 2 3 3 4 5;
 - B. 0 1 2 2 3 4;
 - C. 2 3 4 4 5 6.

```
int a[4][4], i, j;  
for(i=0;i<4;i++) {  
  for(j=0;j<4;j++)  
    a[i][j]=i+j;  
  for(j=i+1;j<4;j++)  
    cout<<a[i][j]<<' ';  
}
```

177. Ce se va afișa în urma executării secvenței de instrucțiuni?
- A. 1 2 2 3 4 5;
 - B. 0 1 2 2 3 4;
 - C. 2 3 4 4 5 6.

```
int a[4][4], i, j;  
for(j=0;j<4;j++)  
  for(i=0;i<4;i++) {  
    a[i][j]=i+j;  
    for(i=j+1;i<4;i++)  
      cout<<a[i][j]<<' ';  
  }
```

178. Ce se va afișa în urma executării secvenței de instrucțiuni?
- A. 36;
 - B. 48;
 - C. 64.

```
int a[4][4], i, j, s=0;  
for(i=0;i<4;i++)  
  for(j=0;j<4;j++) {  
    a[i][j]=i+j;  
    s=s+a[i][j];  
  }  
cout<<s<<endl;
```

179. Ce se va afișa în urma executării secvenței de instrucțiuni?

- A. 8;
- B. 10;
- C. 12.

```
int a[4][4], i, j, s=0;
for(i=0;i<4;i++) {
    for(j=0;j<4;j++)
        a[i][j]=i+j;
    s=s+a[i][i];
}
cout<<s<<endl;
```

180. Fie secvența de instrucțiuni alăturată. Ce se va afișa în urma executării acestei secvențe de instrucțiuni?

- A. 14;
- B. 16;
- C. 18.

```
int a[4][4], i, j, s=0;
for(i=0;i<4;i++) {
    for(j=0;j<4;j++)
        a[i][j]=i+j;
    for(j=i+1;j<4;j++)
        s=s+a[i][j];
}
cout<<s<<endl;
```

181. Ce se va afișa în urma executării secvenței de instrucțiuni?

- A. 14;
- B. 16;
- C. 18.

```
int a[4][4], i, j, s=0;
for(j=0;j<4;j++)
    for(i=0;i<4;i++){
        a[i][j]=i+j;
        for(i=j+1;i<4;i++)
            s=s+a[i][j];
    }
cout<<s<<endl;
```

182. Știind că a este un șir de caractere, iar i și c sunt variabile de tip întreg, identificați prelucrarea realizată de secvența de instrucțiuni alăturată.

- A. eliminarea vocalelor majuscule și minuscule;
- B. numărarea vocalelor majuscule și minuscule;
- C. ordonarea vocalelor în cadrul șirului a.

```
for(i=0;i<strlen(a);i++)
    if(a[i]=='A' || a[i]=='a' ||
       a[i]=='e' || a[i]=='E' ||
       a[i]=='I' || a[i]=='i' ||
       a[i]=='O' || a[i]=='o' ||
       a[i]=='U' || a[i]=='u')
        c++ ;
```

183. Știind că a este un șir de caractere, iar i este o variabilă de tip întreg, ce prelucrare realizează secvența alăturată?

- A. inserarea vocalelor de tip minusculă în șirul a;
- B. eliminarea vocalelor de tip minusculă din șirul a;
- C. ordonarea vocalelor de tip minusculă din șir.

```
char a[20]="AbraCadabra";
for(i=0;i<strlen(a);i++)
    if(a[i]=='a' || a[i]=='e' ||
       a[i]=='i' || a[i]=='o' ||
       a[i]=='u')
        strcpy(a+i,a+i+1);
```

184. Știind că a este un șir de caractere, iar i și k sunt variabile de tip întreg, identificați prelucrarea realizată de secvența de instrucțiuni alăturată.

- A. inserează litere majuscule în șirul a;
- B. numără aparițiile fiecărei litere majuscule în șir;
- C. elimină literele majuscule din șirul a.

```
int b[26]; k=0;
for(char C='A';C<='Z';C++)
{
    b[k]=0;
    for(i=0;i<strlen(a);i++)
        if(C==a[i]) b[k]++;
    k++;
}
```

185. Determinați valoarea expresiei alăturate.

- A. 14;
- B. 2;
- C. 0.

```
}
```

```
((int)fabs(-14.2))%7
```

186. Ce valoare are expresia alăturată?

- A. 36.0;
- B. 40.0;
- C. -28.0.

```
int a=36, b=6, c=3, d=4;  
E = a/b/c*d - a
```

187. Ce se va afișa în urma executării secvenței alăturate de instrucțiuni?

- A. 5 5;
- B. 5 1;
- C. 10 5.

```
int a=10, b=5;  
aux=1;  
if(a<b) aux=a;  
a=b;  
b=aux;  
cout<<a<<' '<<b;
```

188. Care va fi valoarea variabilei întregi x după efectuarea secvenței alăturate?

- A. 8;
- B. 9;
- C. 10.

```
x = 3 + 17 / 3;  
y = x + 1;  
x = y + 1;
```

189. Pentru care dintre perechile de valori de mai jos expresia alăturată are valoarea 1?

- A. 1003 3;
- B. 1100 10;
- C. 1234 12.

```
(a%100==b%100)&&(a>99)||((b>99)
```

190. Ce se va afișa la executarea secvenței de instrucțiuni alăturate?

- A. 2 4 6 8;
- B. 2468;
- C. 24816.

```
p=1;  
do {  
    p*=2;  
    cout<<p;  
} while(p%10!=6);
```

191. Care este valoarea finală pe care o poate lua contorul i pentru ca în urma executării instrucțiunilor alăturate să se afișeze valoarea 4?

- A. 0;
- B. 1;
- C. 3.

```
nr<-0;  
pentru i<-10, ?, -1 execută  
    dacă i mod 2=0 atunci  
        nr<-nr+1  
scrie nr;
```

192. Care este valoarea minimă pe care o poate avea variabila x, astfel încât instrucțiunea cu test final alăturată să efectueze o singură iterație?

- A. 18;
- B. 19;
- C. 20.

```
întreg x,y;  
x←?;  
y←10;  
┌repetă  
│  scrie y;  
│  y←y-1;  
└─ pînă când x>2*y;
```

193. Se consideră algoritmul descris în pseudocod. Ce se va afișa în urma executării algoritmului (operatorul % reprezintă restul la împărțirea întreagă)?

- A. *;
- B. **;
- C. ***.

```
întreg a,b;  
a←10;  
b←2;  
┌cât timp a<15 execută  
│ ┌dacă a%b=0 atunci  
│ │ scrie '*'  
│ └─  
│ a←a-1;  
│ b←b-1;  
└─
```

194. Se consideră algoritmul descris în pseudocod. Ce se va afișa în urma executării instrucțiunilor sale?

- A. 9 3;
- B. 9 4;
- C. 10 4.

```
întreg a,s;  
s←0;  
a←0;  
┌repetă  
│ a←a+1;  
│ s←s+a;  
└─ pînă când s>=10;  
scrie s, a;
```

195. Se consideră algoritmul descris în pseudocod. Care este valoarea maximă pe care o poate avea variabila x, astfel încât instrucțiunea cu test inițială să nu efectueze nici o iterație?

- A. 5;
- B. 6;
- C. 7.

```
întreg x,y;  
x←?;  
y←10;  
┌cât timp 2*x-1<y execută  
│ scrie y;  
│ x←x+1;  
└─
```

196. Ce se va afișa la executarea următoarelor instrucțiuni?

- A. 0 1 2 3 4;
- B. 1 2 3 4 0;
- C. 1 2 3 4.

```
a=1;  
while (a%5!=0) {  
  cout<<a%5;  
  a++;  
}
```

197. Ce valori vor fi afișate în urma rulării secvenței alăturate?

- A. 10 20 30 40 40;
- B. 0 10 20 30 40;
- C. 10 10 20 20 30.

```
int a[5],i;  
for(int i=0;i<5;i++)  
  a[i]=(i+1)*10;  
for(int i=1;i<5;i++)  
  a[i]-=a[i-1];
```



```
for(int i=0;i<5;i++)
  cout<<a[i]<<' ';
```

198. Specificați valoarea elementelor tabloului după execuția instrucțiunilor alăturate.

- A. 2 4 6 8 10;
- B. 0 2 4 6 8;
- C. 2 2 2 2 2.

```
int v[5];
for(i=0;i<5;i++)
  v[i]=2*i;
```

199. Specificați valoarea elementelor tabloului după execuția secvenței alăturate.

- A. 1 1 1 1 1;
- B. 0 1 2 3 4;
- C. 1 2 3 4 5.

```
unsigned char a[5];
for(i=0;i<5;i++)
  a[i]=i;
```

200. Se consideră declarațiile alăturate. Care din referirile elementelor celor doi vectori sunt incorecte?

- A. x[y[1]];
- B. x[1];
- C. y(x[1]).

```
char x[301];
unsigned char y[301],i;
```

201. Ce se va afișa la executarea secvenței de instrucțiuni?

- A. 121 11;
- B. 121 12;
- C. 120 12.

```
a=10; b=a+1;
if(a!=b) b++;
else a++;
a=b*a;
cout<<a<<' '<<b;
```

202. Considerăm un tablou cu n linii și m coloane. Câte elemente sunt situate pe marginea tabloului (prima și ultima linie, prima și ultima coloană)?

- A. $2*m+2*n$;
- B. $2*n+2*m-4$;
- C. $2*n+2*m-2$.

203. Considerăm un tablou cu n linii și m coloane în care toate elementele primei linii sunt nule. Știind că nu există alt element egal cu zero (nesituat pe prima linie), câte elemente nenule sunt în tablou?

- A. $(m+n)-n$;
- B. $(m+n)-m$;
- C. $m*n-n$.

204. Considerăm un tablou cu n linii și m coloane în care toate elementele primei coloane sunt nule. Știind că nu există alt element egal cu zero (nesituat pe prima coloana), câte elemente nenule sunt în tablou?

- A. $(m+n)-n$;
- B. $(m+n)-m$;
- C. $m*n-n$.

205. Specificați care va fi conținutul tabloului a[3][3] după execuția secvenței de instrucțiuni alăturate.

- A. 2 2 2 B. 3 4 5 C. 2 3 4
4 4 4 5 6 7 3 4 5
6 6 6 7 8 8 4 5 6

```
int a[3][3];
for(i=0;i<3;i++)
for(j=0;j<3;j++)
a[i][j]=i+j+2;
```

206. Specificați care va fi conținutul tabloului a[3][3] după execuția secvenței de instrucțiuni alăturate.

- A. 2 2 2 B. 3 4 5 C. 2 2 2
4 4 4 5 6 7 3 3 3
6 6 6 7 8 8 4 4 4

```
for(i=0;i<3;i++)
for(j=0;j<3;j++)
a[i][j]=2*(i+1)+j+1;
```

207. Considerăm o matrice pătratică cu n linii și n coloane. Dacă notăm cu i și j indicii de linie, respectiv de coloană al unui element din matrice, care este condiția ca un element să se regăsească pe diagonala principală a matricii?

- A. $j=i$;
B. $j=n-i$;
C. $j=i-n$.

208. Considerăm o matrice pătratică cu n linii și n coloane. Dacă notăm cu i și j indicii de linie, respectiv de coloană al unui element din matrice, care este condiția ca un element să se regăsească pe diagonala secundară a matricii?

- A. $j=i$;
B. $j=n-i$;
C. $j=i-n$.

209. Ce se afișează în urma executării secvenței de program alăturate?

- A. Masina;
B. Macara;
C. MasinaMacara.

```
x="Masina"; y="Macara";
if(strcmp(x,y)>0)
cout<<x;
else
if(strcmp(x,y)<0)
cout<<y;
else
cout<<"Incorect";
```

210. Ce se va afișa în urma executării secvenței de program alăturate?

- A. MacarA;
B. AracaM;
C. ArcaaaM.

```
x="MacarA";
for(i=0;i<strlen(s)/2;i++) {
char x=s[i];
s[i]=s[strlen(s)-i-1];
s[strlen(s)-i-1]=x;
}
```

211. Ce se va afișa în urma executării secvenței de program alăturate?

- A. MacarA;
B. AracaM;
C. ArcaaaM.

```
x="MacarA";
for(i=0;i<strlen(s);i++) {
char x=s[i];
s[i]=s[strlen(s)-i-1];
s[strlen(s)-i-1]=x;
}
```

212. Știind că a este un șir de caractere și i o variabilă de tip întreg, identificați prelucrarea realizată asupra șirului a.

- A. transformarea caracterelor de tip minusculă în majuscula corespunzătoare;
- B. transformarea caracterelor de tip majusculă în minuscula corespunzătoare;
- C. ordonarea alfabetică a majusculilor în cadrul șirului.

```
for(i=0;i<strlen(a);i++)
if(a[i]>='A' && a[i]<='Z')
a[i]+=32;
```

213. Ce se va afișa în urma executării secvenței de program alăturate?

- A. Pramare;
- B. gramare;
- C. are.

```
a="Programare";
for(i=1;i<3;i++)
strcpy(a+1,a+2);
cout<<a;
```

214. Specificați ce se va afișa după execuția secvenței alăturate, pentru n=5.

- | | | |
|--------------|--------------|--------------|
| A. 1 1 1 1 1 | B. 1 2 3 4 5 | C. 1 2 3 4 5 |
| 2 2 2 2 2 | 1 2 3 4 5 | 2 3 4 5 1 |
| 3 3 3 3 3 | 1 2 3 4 5 | 3 4 5 1 2 |
| 4 4 4 4 4 | 1 2 3 4 5 | 4 5 1 2 3 |
| 5 5 5 5 5 | 1 2 3 4 5 | 5 4 3 2 1 |

```
int i,j;
for(i=1;i<=n;i++){
for(j=1;j<=n;j++){
cout<<i<<" ";
}
cout<<"\n";
}
```

215. Specificați ce se va afișa după execuția secvenței alăturate, pentru n=5.

- | | | |
|--------------|--------------|--------------|
| A. 1 1 1 1 1 | B. 1 2 3 4 5 | C. 1 2 3 4 5 |
| 2 2 2 2 2 | 1 2 3 4 5 | 2 3 4 5 1 |
| 3 3 3 3 3 | 1 2 3 4 5 | 3 4 5 1 2 |
| 4 4 4 4 4 | 1 2 3 4 5 | 4 5 1 2 3 |
| 5 5 5 5 5 | 1 2 3 4 5 | 5 4 3 2 1 |

```
int i,j;
for(i=1;i<=n;i++){
for(j=1;j<=i;j++){
cout<<j<<" ";
}
cout<<"\n";
}
```

216. Specificați ce se va afișa după execuția secvenței alăturate, pentru n=5.

- | | | |
|--------------|-----------|-----------|
| A. 1 1 1 1 1 | B. 1 | C. 1 |
| 2 2 2 2 2 | 1 2 | 2 2 |
| 3 3 3 3 3 | 1 2 3 | 3 3 3 |
| 4 4 4 4 4 | 1 2 3 4 | 4 4 4 4 |
| 5 5 5 5 5 | 1 2 3 4 5 | 5 5 5 5 5 |

```
int i,j;
for(i=1;i<=n;i++){
for(j=1;j<=i;j++){
cout<<i<<" ";
}
cout<<"\n";
}
```

217. Specificați ce se va afișa după execuția secvenței alăturate, pentru n=5.

- | | | |
|--------------|-----------|--------------|
| A. 1 1 1 1 1 | B. 1 | C. 1 2 3 4 5 |
| 2 2 2 2 2 | 1 2 | 2 3 4 5 1 |
| 3 3 3 3 3 | 1 2 3 | 3 4 5 1 2 |
| 4 4 4 4 4 | 1 2 3 4 | 4 5 1 2 3 |
| 5 5 5 5 5 | 1 2 3 4 5 | 5 4 3 2 1 |

```
int i,j;
for(i=1;i<=n;i++){
for(j=1;j<=n;j++){
cout<<j<<" ";
}
cout<<"\n";
}
```

218. Specificați ce se va afișa după execuția secvenței alăturate, pentru n=5.

- | | | |
|--------------|--------------|--------------|
| A. 1 1 1 1 1 | B. 1 2 3 4 5 | C. 1 2 3 4 5 |
| 2 2 2 2 2 | 1 2 3 4 5 | 2 3 4 5 1 |
| 3 3 3 3 3 | 1 2 3 4 5 | 3 4 5 1 2 |
| 4 4 4 4 4 | 1 2 3 4 5 | 4 5 1 2 3 |
| 5 5 5 5 5 | 1 2 3 4 5 | 5 4 3 2 1 |

```
int i,j;
for(i=1;i<=n;i++){
  for(j=1;j<=n;j++){
    if(i+j>n+1)
      cout<<i+j-n-1<<" ";
    else
      cout<<i+j-1<<" ";
  }
  cout<<"\n";
}
```

219. Care din următoarele secvențe de program produc același rezultat ca și secvența alăturată?

- | | |
|--|---|
| A. int i=100,n=100;
int sum=0;
do{ sum+=i++;
}while(i<n);
cout<<sum;); | B. int i=100,n=100;
int sum=0;
while(i<n){
sum+=i++;};
cout<<sum;); |
|--|---|

```
int i, n=100, sum=0;
for(i=100;i<n;sum+=i++);
cout<<sum;
```

C. Nici una din variante

220. Care operatori au cea mai mare precedență în evaluarea unei expresii aritmetice?

- A. operatorii unari;
- B. operatorii binari *, /;
- C. operatorii binari +, -.

221. Care din următoarele tipuri de date sunt tipuri structurate?

- A. tipuri înregi;
- B. tipuri reale;
- C. tipul tablou.

222. Care afirmații privind secvența alăturată sunt adevărate?

- A. se calculează valoarea lui f la puterea num;
- B. se calculează valoarea lui num!;
- C. se calculează suma 1 + ... + num.

```
int main(){
  int i; long f=1;
  for(i=1;i<=num; i++)
    f = f * i;
}
```

223. Care afirmații privind secvența alăturată sunt adevărate?

- A. se vor afișa de 10 ori valorile lui x și y;
- B. se va afișa o singură dată valoarea lui x și a lui y;
- C. caracterul ; aflat după instrucțiunea de decizie if generează eroare.

```
int x=10, y=100%90, i;
for(i=1;i<10; i++)
  if(x != y);
cout<<"x = "<<x<<"\n";
cout<<"y = "<<y<<"\n";
```

224. Care din următoarele operații este incorectă?

- A. $\text{int } a = 3; a = a;$
- B. $\text{float } a = 3.14; a = a\%3;$
- C. $\text{int } a = 3; a = a++.$

225. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $a=12$ și $b=18$?

- A. 12 18;
- B. 40 0;
- C. 18 12.

```

citește a, b
 $a \leftarrow a+b$ 
 $a \leftarrow a-b$ 
 $b \leftarrow a-b$ 
scrie a ; scrie b

```

226. Se consideră programul pseudocod alăturat, cu a și b numere naturale nenule. Ce se va afișa pentru $a=12$ și $b=18$?

- A. 18;
- B. 6;
- C. 12.

```

citește a, b
cât timp  $b \neq 0$  execută
┌─dacă  $a < b$ 
│   ┌─atunci  $a \leftarrow a+b$ 
│   │    $a \leftarrow a-b$ 
│   │    $b \leftarrow a-b$ 
│   └─
│   └─
└─┬─
   └─ $a \leftarrow a-b$ 
scrie a

```

227. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $n=6$?

- A. 1 2 3 4 5 6;
- B. 1 2 4 6 8 12;
- C. 1 2 3 5 8 13.

```

citește n {număr natural nenul}
 $i \leftarrow 1$ 
 $j \leftarrow 1$ 
cât timp  $n > 0$  execută
┌─scrie i
│    $i \leftarrow i+j$ 
│    $j \leftarrow i-j$ 
│    $n \leftarrow n-1$ 
└─

```

228. Se consideră programul pseudocod alăturat, unde s-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y. Ce se va afișa pentru $x=3$ și $y=2014$?

- A. 9;
- B. 7;
- C. 3.

```

citește x, y {numere naturale nenule}
 $\text{rez} \leftarrow 1$ 
cât timp  $y > 0$  execută
┌─ $\text{rez} \leftarrow \text{rez} * x \% 10$ 
│    $y \leftarrow y-1$ 
└─
scrie rez

```

229. Se consideră programul pseudocod alăturat.
Ce se va afișa pentru $n=11$?
- A. 1024;
 - B. 2048;
 - C. 4096.

```
citește n {număr natural nenul}
x ← 1
sum ← 1
cât timp n > 0 execută
    sum ← sum + x
    x ← 2 * x
    n ← n - 1
scrie sum
```

230. Se consideră programul pseudocod alăturat, unde s-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y . Ce se va afișa pentru $n=6$?
- A. Nr. perfect;
 - B. Nr. abundent;
 - C. Nr. deficient.

```
citește n {număr natural nenul}
sum ← 0
i ← 1
cât timp i < n + 1 execută
    dacă n % i == 0 atunci
        sum ← sum + i
    i ← i + 1
dacă sum == 2 * n atunci
    scrie "Nr. perfect"
altfel
    dacă sum < 2 * n atunci
        scrie "Nr. deficient"
    altfel
        scrie "Nr. abundent"
```

231. Proprietatea unui algoritm conform căreia, la fiecare moment operația care urmează a fi executată este unic determinată, definită și realizabilă, se numește:
- A. claritate;
 - B. generalitate;
 - C. finitudine.
232. Structura repetitivă al cărei corp se execută cel puțin o dată, indiferent de îndeplinirea sau nu a expresiei condiționale este:
- A. structura repetitivă cu contor;
 - B. structura repetitivă condiționată posterior;
 - C. structura repetitivă condiționată anterior.
233. Structura care se definește prin selecția între două sau mai multe acțiuni este:
- A. structura secvențială;
 - B. structura alternativă;
 - C. structura repetitivă.

234. Se consideră programul pseudocod alăturat, unde s-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y . Ce se va afișa pentru $n=5$?

- A. Nr. perfect;
- B. Nr. abundent;
- C. Nr. deficient.

```

citește n {număr natural nenul}
sum ← 0
i ← 1
cât timp i < n+1 execută
    dacă n%i == 0 atunci
        sum ← sum+i
    i ← i+1
dacă sum == 2*n atunci
    scrie "Nr. perfect"
altfel
    dacă sum < 2*n atunci
        scrie "Nr. deficient"
    altfel
        scrie "Nr. abundent"

```

235. Se consideră programul pseudocod alăturat, unde s-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y . Ce se va afișa pentru $n=12$?

- A. Nr. perfect;
- B. Nr. abundent;
- C. Nr. deficient.

```

citește n {număr natural nenul}
sum ← 0
i ← 1
cât timp i < n+1 execută
    dacă n%i == 0 atunci
        sum ← sum+i
    i ← i+1
dacă sum == 2*n atunci
    scrie "Nr. perfect"
altfel
    dacă sum < 2*n atunci
        scrie "Nr. deficient"
    altfel
        scrie "Nr. abundent"

```

236. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $n=1231$? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. 1321;
- B. 1123;
- C. 1231.

```

citește n {număr natural nenul}
m ← 0
cât timp n > 0 execută
    m ← m*10+n%10
    n ← [n/10]
scrie m

```

237. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $i=11, j=60, k=61$?

- A. Sunt pitagoreice;
- B. Nu sunt pitagoreice;
- C. Niciuna din variante.

```

citește i,j,k
{numere naturale,  $i < j < k$ }
dacă  $i*i+j*j==k*k$  atunci
    scrie "Sunt pitagoreice"
altfel
    scrie "Nu sunt pitagoreice"

```

238. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $a=220, b=284$? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. Sunt prietene;
- B. Nu sunt prietene;
- C. Niciuna din variante.

```

citește a, b {numere naturale nenule}
sumA ← 0 ; i ← 1
cât timp  $i < [a/2] + 1$ 
    execută
        dacă  $a\%i == 0$  atunci
            sumA ← sumA + i
        i ← i + 1
sumB ← 0
i ← 1
cât timp  $i < [b/2] + 1$ 
    execută
        dacă  $b\%i == 0$  atunci
            sumB ← sumB + i
        i ← i + 1
dacă sumA == b și sumB == a
    atunci
        scrie "Sunt prietene"
    altfel
        scrie "Nu sunt prietene"

```

239. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $n=1231$? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. 1;
- B. 2;
- C. 3.

```

citește n {număr natural nenul}
max ← -1
cât timp  $n > 0$  execută
    dacă  $max < n\%10$  atunci
        max ←  $n\%10$ 
    n ←  $[n/10]$ 
scrie max

```


240. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $a=3$, $b=4$? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. Sunt prietene;
- B. Nu sunt prietene;
- C. Nimic.

```

citește a, b {numere naturale nenule}
sumA ← 0 ; i ← 1
cât timp i < [a/2]+1
    execută
        dacă a%i == 0 atunci
            sumA ← sumA+i
        i ← i+1
sumB ← 0 ; i ← 1
cât timp i < [b/2]+1
    execută
        dacă b%i == 0 atunci
            sumB ← sumB+i
        i ← i+1
dacă sumA == b și sumB == a
    atunci
        scrie "Sunt prietene"
    altfel
        scrie "Nu sunt prietene"
    
```

241. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $n=3668$, $m=863$? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. Sunt asemenea;
- B. Nu sunt asemenea;
- C. Nimic.

```

citește m, n {numere naturale nenule}
pentru i=0,9 execută
    cifre[i] ← 0
cât timp n > 0 execută
    cifre[n%10] ← 1
    n ← [n/10]
cât timp m > 0 execută
    cifre[m%10] ← cifre[m%10]-1
    m ← [m/10]
msg ← "Sunt asemenea"
pentru i=0,9 execută
    dacă cifre[i] <> 0 atunci
        msg ← "Nu sunt asemenea"
        break;
scrie msg
    
```

242. Care este rezultatul evaluării expresiei?
A. 104; B. 34; C. 94

15+5*5-10+100 / (20+5)

243. Ce valoare va avea variabila b după rularea codului alăturat?

- A. 0;
- B. 1;
- C. Va fi generată o eroare.

```
int b;  
b=(1 || false)&&(1+0);
```

244. Ce valoare va avea variabila b după rularea codului alăturat?

- A. 0;
- B. 1;
- C. Va fi generată o eroare.

```
int b;  
b=(0 || false)&&(1+0);
```

245. Ce valoare va avea variabila x la sfârșitul execuției codului alăturat?

- A. 103;
- B. 43643;
- C. 43.

```
int x;  
x=4360;  
x=x%10*100+43;
```

246. Ce valoare va avea variabila x la sfârșitul execuției codului alăturat?

- A. 4349;
- B. 479;
- C. 4403.

```
int x;  
x=436;  
x=x%10+x/10*100+43;
```

247. Se consideră următorul algoritm descris în pseudocod. S-a notat cu $x \% y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z . Ce se va afișa pentru $n=10021133$?

- A. 31201;
- B. 10213;
- C. 33112001.

```
citește n {număr natural nenul}  
x ← 0 ; p ← 1  
cât timp n <> 0 execută  
  x ← x + n % 10 * p  
  dacă n % 10 = n % 100  
  │ n ← [n / 100]  
  │ ■  
  altfel  
  │ n ← [n / 10]  
  │ ■  
  p ← p * 10  
  │ ■  
scrie x
```

248. Care dintre următorii algoritmi pseudocod este corect, dacă n natural nenul citit de la tastatură este 78623 și la final este afișat rezultatul 31347. S-a notat cu $x \% y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

A. _____ B. _____ C. _____

```

citește n
x ← 0
cât timp n > 0 execută
  dacă n%10%2=0
    x ← x*10+n%10/2
  altfel
    x ← x*10+n%10
  n ← [n/10]
scrie x

```

```

citește n
x ← 0
cât timp n > 0 execută
  dacă n%10/2=0
    x ← x*10+(n%10)%2
  altfel
    x ← x*10+n%10
  n ← [n/10]
scrie x

```

```

citește n
x ← 0 ; p ← 1
cât timp n > 0 execută
  dacă n%10/2=0
    x ← x+((n%10)%2)*p
  altfel
    x ← x*10+n%10
  n ← [n/10]
  p ← p*10
scrie x

```

249. Se consideră următorul algoritm în pseudocod. S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z . Ce va realiza acest cod?

- A. scrie invers numărul n și elimina cifrele pare;
- B. elimină din numărul n cifrele divizibile cu 2;
- C. elimină din numărul introdus cifrele 2.

```

citește n {număr natural nenul}
x ← 0 ; p ← 1
cât timp n > 0 execută
  dacă (n%10)%2 != 0 sau
  (n%10) != 2
    x ← x+n%10*p
    p ← p*10
  n ← [n/10]
scrie x

```

250. Se consideră algoritmul pseudocod alăturat, care înlocuiește cu 0 cifrele pare a unui număr n introdus de la tastatura. S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z . Cu ce pot fi înlocuite punctele de suspensie?

- A. $(n\%10)/2=0$;
- B. $(n\%10)\%2!=0$;
- C. $(n\%10)\%2=0$.

```

citește n {număr natural nenul}
x ← 0
p ← 1
cât timp n > 0 execută
  dacă ...
    x ← x+n%10*p
  p ← p*10
  n ← [n/10]
scrie x

```

251. Ce va realiza următorul algoritm în pseudocod? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y și cu $[z]$ partea întreagă a numărului real z .

- A. afișează pentru un număr dat, cea mai mare cifră din primele 4 cifre ale numărului;
- B. afișează pentru un număr dat, cea mai mare cifră strict mai mică decât 5;
- C. afișează pentru un număr dat, cea mai mare cifră, numai dacă numărul conține cifre mai mici ca 5.

```

citește n {număr natural nenul}
m ← 0
cât timp n > 0 execută
  dacă n%10 > m și n%10 < 5
    m ← n%10
  n ← [n/10]
scrie m

```

252. Dacă se consideră $n=5$ și $x=2963$, care va fi valoarea variabilei c afișată de algoritmul în pseudocod alăturat? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y .

- A. 1;
- B. 2;
- C. 3.

```
n ← 5 ; x ← 2963 ; c ← 0
cât timp n > 0 execută
    dacă x % 3 = 0
        c ← c + 1
    n ← n - 1
    x ← x + 1
scrie c
```

253. Se consideră următorul algoritm descris în pseudocod. Cu ce secvență se pot înlocui punctele de suspensie, astfel încât după execuția algoritmului, variabila c să aibă valoarea 2? S-a notat cu $x\%y$ restul împărțirii numărului întreg x la numărul întreg y .

```
n ← 4 ; x ← 48 ; c ← 0 ; i ← 0
cât timp n >= 0 execută
    dacă x % 2 = 0
        c ← c - 1
    ...
scrie c
```

A.

```
n ← n - 2
i ← i + 1
x ← x + i
c ← c + 2
```

B.

```
n ← n - i
i ← i + 2
x ← x + i
c ← c + 2
```

C.

```
n ← n - 1
i ← i + 1
x ← x + i
c ← c + 1
```

254. Se consideră programul alăturat. Care va fi valoarea variabilei x după execuția acestui program?

- A. 7;
- B. 8;
- C. 9.

```
int x=0, n=7, i;
while(n>0) {
    for(i=0; i<3; i++) {
        x++;
        n--;
    }
}
```

255. Care este rezultatul afișat de programul alăturat, pentru $x=5$?

- A. 17;
- B. 19;
- C. 5.

```
int x, i, j;
cin >> x;
for(i=0; i<=3 ; i++) {
    x=x-1;
    for(j=0; j<=1; j++) {
        x=x+2;
    }
}
cout << x;
```

256. Se dă programul alăturat. Cu ce linie de cod se pot înlocui punctele de suspensie, astfel încât la sfârșitul execuției programului, variabila x să aibă valoarea 16?

- A. `n>=-1;`
- B. `n==0;`
- C. `n>=0;`

```
int x, n, i;
x=0; n=3;
do {
    n--;
    for(i=0;i<2; i++) {
        x=x+2;
    }
}
while (...);
cout << x;
```

257. Ce valori vor fi afișate în urma rulării secvenței de program alăturate?

- A. 0 1 4 9 16 5 6;
- B. 0 1 4 9 4 5 6;
- C. 0 1 4 9 4 5.

```
int a[10], i;
for(i=0;i<6;i++){
    if(i<4) {
        a[i]=i*i;
    }
    else {
        a[i]=1+3;
    }
    cout << a[i];
}
```

258. Ce face următoarea secvență de program?

- A. Face suma tablourilor a și b termen cu termen și rezultatele sunt depuse în tabloul c;
- B. Creează tabloul c format din termenii tabloului a urmați apoi de termenii tabloului b ;
- C. Creează tabloul c format din termenii tabloului a urmați apoi de suma termenilor tablourilor a și b termen cu termen.

```
int a[5]={3,5,1,2,4};
int b[6]={5,7,3,9,8,1};
int c[11]={0,0,0,0,0,0,0,0,0,0,0};
int i, j;
for(i=0;i<5;i++) {
    c[i]=a[i];
}
for(j=0;j<6;j++) {
    c[j]=c[j]+b[i];
}
```

259. Ce valoare va avea tabloul v în urma rulării secvenței de program alăturate?

- A. 1 2 4 7 11 16 1 3 6 10;
- B. 1 2 4 7 11 0 1 3 6 10;
- C. 1 2 4 7 11 1 3 6 10 0.

```
int v[10], i;
v[0]=1;
for(i=1;i<6;i++){
    v[i]=v[i-1]+i;
}
i=1;
while(i<=5){
    v[i+5] = v[i]-1; i++;
}
```

260. Cu ce linie de cod se pot înlocui punctele de suspensie, astfel încât vectorul v să aibă după rularea secvenței alăturate valorile 3 6 10 0 0 0 0 0 0?

- A. $v[j]=v[i]+j$;
- B. $v[i]=v[i]+j$;
- C. $v[i]=v[1]+3$.

```
int v[9]={0,0,0,0,0,0,0,0,0};
int i, j;
for(i=0;i<3;i++){
    for(j=1;j<i+3;j++){
        ...
    }
}
```

261. Care dintre instrucțiunile de mai jos generează o eroare, dacă vectorul v se inițializează cu valorile alăturate?

- A. $v[2] = 1.3$;
- B. $v[1] = "8"$;
- C. $v[1] = 'a'$.

```
int v[3]={0,2,5};
```

262. Ce valoare va avea matricea m în urma rulării secvenței de program alăturate?

- A.
1 2 3
2 4 6
3 6 9

- B.
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16

```
int m[4][4], i, j;
for(i=0;i<4;i++) {
    for(j=0;j<4;j++) {
        m[i][j]=(i+1)*(j+1);
    }
}
```

- C.
2 3 4 5
3 4 5 6
4 5 6 7
5 6 7 8

263. Ce rezultă în urma rulării secvenței alăturate?

- A. în variabila s se păstrează suma elementelor de pe ultimele două linii ale matricei m;
- B. în variabila s se păstrează suma elementelor de pe ultimele trei linii ale matricei m;
- C. în variabila s se păstrează suma elementelor de pe ultimele două linii a matricei m, din care se scad valorile elementelor de pe ultima coloană a matricei m.

```
int m[4][4]={{1,2,3,4},
{3,4,5,6},{8,7,6,5},{1,2,4,6}};
int i, j, s=0;
for(i=0;i<4;i++) {
    for(j=0;j<4;j++)
        if(i>1)
            s=s+m[i][j];
        else
            if(j==4)
                s=s-m[i][j];
}
```

264. Câte elemente vor fi inițializate în matricea t, folosind secvența de program alăturată, dacă considerăm $m=3$ și $n=4$?

- A. 12;
- B. 16;
- C. 9.

```
for(i=0;i<=m;i++)
    for(j=1;j<n;j++)
        t[i][j-1]=(i*i)+j;
```

265. Care dintre secvențele de cod de mai jos calculează în variabila s suma tuturor elementelor de pe liniile impare ale matricei m?

A.

```
for(i=0;i<m;i=i+2)
for(j=0;j<n;j++)
if(j%2!=0)
s+=t[i][j];
```

B.

```
for(i=0;i<m;i++)
if(i%2!=0) {
j=0;
for(;j<n;j++)
s=t[i][j];
}
```

C.

```
i=0;
for(;i<m;i++){
j=0;
for(;j<n;j++)
if(i%2!=0)
s+=t[i][j];
}
```

266. Ce set de linii de cod pot înlocui punctele de suspensie din secvență de cod alăturată, astfel încât după rularea secvenței să se rețină în s suma elementelor de pe diagonala principală a matricei t?

```
for(i=0;i<n;i++)
for(j=0;j<=i;j++) {
...
}
```

A.

```
if(j<=i) {
s+=t[i][j];
}
```

B.

```
if(j<i) {
s+=t[i][j];
}
```

C.

```
if(j==i) {
s+=t[i][j];
}
```

267. Se consideră secvența de cod alăturată pentru sortarea tablourilor. Dacă tabloul a este format din elementele {1, 3, 6, 2, 4}, care este valoarea variabilei c afișată după rularea secvenței.

- A. 2;
- B. 3;
- C. 4.

```
c=0;
do {
ok=false;
for(i=0;i<5;i++) {
if(a[i] > a[i+1]) {
ok=true;
aux=a[i];
a[i]=a[i+1];
a[i+1]=aux;
c++;
}
}
} while(ok);
cout << c;
```

268. Se consideră secvența de cod de căutare binară de mai jos. Căutarea se face în tabloul *t*, inițializat și ordonat anterior, iar valoarea căutată este reținută în variabila *caut*. La finalul rulării algoritmului, în variabila *pozitie* se reține valoarea indicelui din tabloul *t* unde a fost găsită valoarea căutată *caut*, sau -1 dacă nu a fost găsită valoarea căutată în tabloul *t*.

Ce linie trebuie modificată pentru ca algoritmul să funcționeze corect?

- A. Linia : `while (st < dr && !gasit) {`;
- B. Linia : `if (t[mijloc] < caut) {`;
- C. Nici o linie. Algoritmul este corect.

```
pozitie=-1;
st=0;
//valoarea primului indice al tabloului
dr = sizeof(t)/sizeof(t[0])-1;
//valoarea ultimului indice al tabloului
gasit=false;
while (st <= dr && !gasit) {
    mijloc = (st + dr) / 2;
    if(t[mijloc] == caut) {
        gasit = true;
        pozitie = mijloc;
    }
    else
        if (t[mijloc] > caut)
            st = mijloc + 1;
        else
            dr = mijloc - 1;
}
```

269. Ce valori vor avea elementele tabloului *t* după rularea secvențe de cod, dacă se știe că vectorul *t* avea înainte de execuție următoarele valori: {1, 41, 5, 9, 23, 36, 15, 91}

- A. {1, 41, 5, 11, 23, 38, 17, 91};
- B. {3, 43, 7, 9, 25, 36, 15, 93};
- C. {3, 42, 6, 9, 24, 36, 15, 93}.

```
for(j=0;j<2;j++) {
    for(i=0;i<8;i++) {
        if(t[i]%3!=0) {
            t[i]=t[i]+1;
        }
    }
}
```

270. Ce valori va conține tabloul *c* după rularea secvenței de program alăturată?

- A. {1, 2, 3, 7, 9, 10, 14, 16};
- B. {3, 7, 9, 14, 1, 2, 10, 16}.
- C. {3, 1, 7, 2, 9, 10, 14, 16}.

```
int a[4]={3,7,9,14};
int b[4]={1,2,10,16}, c[8];
while(k < 8) {
    k++;
    if((j==4)||((a[i]<b[j]))&&(i!=4)){
        c[k]=a[i];
        i++;
    }
    else {
        c[k] = b[j];
        j++;
    }
}
```


271. Ce operație face secvența de cod alăturată?
- Ordonează crescător prima jumătate a tabloului v;
 - Ordonează descrescător elementele tabloului v;
 - Inversează ordinea elementelor din tabloul v.

```
int i, j, t, aux; bool ok;
int v[7]={3,8,6,1,4,11,9};
t=7/2;
do {
    ok=true;
    for(i=0;i<t; i++) {
        if(i == (t-1))
            ok=false;
        aux=v[i];
        v[i]=v[(7-1)-i];
        v[(7-1)-i]=aux;
    }
} while(ok);
```

272. Dacă variabilele x și y sunt de tip int, ce valori vor avea acestea la finalul executării secvenței alăturate?
- x=0 și y=0;
 - x=0 și y=30;
 - x=0 și y=55.

```
x=5;
y=0;
do {
    x=x-1;
    y=y+x*x;}
while(x!=0);
```

273. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural y și cu $[z]$ partea întreagă a numărului real z. Ce se va afișa pentru $n=102206$?
- 897793;
 - 786682;
 - 397798.

```
citește n {număr natural}
x ← 0 ; p ← 1
repetă
    x ← x + (9 - n % 10) * p
    n ← [n / 10]
    p ← p * 10
până când n = 0
scrie x
```

274. Pentru care din următoarele valori ale variabilei n secvența de program alăturată afișează valoarea 0 în urma executării ei.
- 1111;
 - 9282;
 - 3003.

```
while(n%10 >= 2)
    n = n / 10;
cout << n;
```

275. Se consideră programul pseudocod alăturat. S-a notat cu $x \bmod y$ restul împărțirii lui x la y și cu $x \operatorname{div} y$ câtul împărțirii lui x la y . Ce se va afișa pentru $n=27155$?

- A. 7;
- B. 2;
- C. 5.

```

citește n {n nr. natural}
k ← 0
pentru i ← 0,9 execută
    x ← n
    cât timp x > 0 execută
        dacă x mod 10 = i atunci
            k ← i
        x ← x div 10
serie k

```

276. Stabiliți care dintre următoarele expresii este adevărată dacă și numai dacă numărul întreg x este par și strict pozitiv.

- A. $!((x \% 2 != 0) \&\& (x < 0))$;
- B. $(x \% 2 == 1) \&\& (x > 0)$;
- C. $!((x \% 2 != 0) || (x <= 0))$.

277. Fie v un tablou unidimensional format din n numere întregi ($n \geq 2$). Știind că primul element din tablou se află pe poziția 1, indicați care dintre următoarele secvențe atribuie variabilei întregi \max cea mai mare valoare din tabloul v .

A.

```

for(i=2; i<=n; i++)
    if(v[i-1] <= v[i])
        v[i-1] = v[i];
max = v[1];

```

B.

```

for(i=1; i<=n-1; i++)
    if(v[i+1] > v[i])
        v[i] = v[i+1];
max = v[1];

```

C.

```

for(i=2; i<=n; i++)
    if(v[i-1] > v[i]) v[i] = v[i-1];
max = v[n];

```

278. Se consideră programul pseudocod alăturat. S-a notat cu $x \bmod y$ restul împărțirii lui x la y și cu $x \operatorname{div} y$ câtul împărțirii lui x la y . Ce se va afișa pentru $n=4840$?

- A. 11;
- B. 13;
- C. 15.

```

citește n {n ∈ N, n ≥ 2}
k ← 2 ; t ← n
cât timp t ≠ 1 execută
    cât timp t mod k = 0 execută
        m ← k
        t ← t div k
    k ← k + 1
serie m

```

279. Stabiliți care dintre următoarele expresii este adevărată dacă și numai dacă numerele reale strict pozitive a, b, c reprezintă lungimile laturilor unui triunghi dreptunghic.

- A. $(c * c == a * a + b * b) || (a * a == b * b + c * c) || (b * b == a * a + c * c)$;
- B. $(c * c == a * a + b * b) \&\& (c >= b >= a)$;
- C. $(c * c == a * a + b * b) \&\& (a * a == b * b + c * c) \&\& (b * b == a * a + c * c)$.

280. După executarea secvenței de program alăturate variabilele x și y de tip int vor avea valorile:

- A. x=3 y=3;
- B. x=5 y=3;
- C. x=3 y=5.

```
x=1;
y=7;
do{
    x++; y--;
}while(x<=y);
```

281. Se consideră programul pseudocod alăturat. S-a notat cu x mod y restul împărțirii lui x la y și cu x div y câtul împărțirii lui x la y.

Ce se va afișa pentru n=1764?

- A. 1;
- B. 7;
- C. 6.

```
citește n {n∈N,n≥2}
k←1
d←2
repetă
    p←0
    cât timp n mod d=0 execută
        p←p+1
        n←n div d
    ■
    dacă p mod 2=1 atunci
        k←0
    ■
    d←d+1
până când (n=1)
scrie k
```

282. Ce valori vor avea variabilele de tip întreg x și y după executarea secvenței alăturate?

- A. x=5 y=7;
- B. x=7 y=5;
- C. x=6 y=5.

```
x=1;
y=11;
while(x<=y){
    x=x+1;
    y=y-1;
}
```

283. Fie v un tablou unidimensional format din n componente întregi, numerotate de la 1 la n, iar p și k două variabile de tip întreg. În care dintre următoarele secvențe variabilei p i se atribuie prima valoare strict pozitivă din tabloul v, dacă există o astfel de valoare, sau 0 în caz contrar?

A.

```
p=0;k=1;
while(v[k]>=0) k++;
p=v[k];
```

B.

```
p=0;k=n;
while(v[k]<=0) k--;
p=v[k];
```

C.

```
p=0;
for(k=1;k<=n;k++)
    if((v[k]>0)&&(p==0))
        p=v[k];
```

284. Care din secvențele de program pseudocod de mai jos elimină corect elementul x₁ din vectorul cu componentele x₁, x₂,..., x_n.

A.

```
i←1
cât timp i<=n-2 execută
    xi←xi+1; i←i+1
    ■
n←n-1
```

B.

```
i←2
cât timp i<=n execută
    xi←xi-1; i←i+1
    ■
n←n-1
```

C.

```
i←2
cât timp i<=n execută
    xi-1←xi; i←i+1
    ■
n←n-1
```

285. Ce reprezintă rezultatul afișat de programul pseudocod?

- A. suma numerelor naturale impare mai mici sau egale decât n ;
- B. suma primelor n numere naturale;
- C. suma numerelor naturale impare mai mici decât n .

```
citește n (n>0 nr natural)
s←0; k←1
┌cât timp k≤n execută
│ s←s+k; k←k+2
└─┘
scrie s
```

286. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii lui x la y și cu $[x]$ partea întreagă a numărului real x . Care este valoarea afișată pentru $n=83425$ și $k=3$?

- A. 6;
- B. 7;
- C. 9.

```
citește n,k
(n,k numere naturale)
s←0
┌cât timp n>0 și k>0
│ execută
│   c←n%10
│   dacă c%2=0 atunci
│     ┌ s←s+c
│     └─┘
│   n←[n/10]; k←k-1
└─┘
scrie s
```

287. Stabiliți care dintre următoarele expresii logice are valoarea ADEVĂRAT dacă și numai dacă valoarea variabilei reale x se găsește în afara intervalului $[0, 1]$.

- A. $x < 0$ și $x > 1$;
- B. $x < 0$ sau $x > 1$;
- C. $x \geq 0$ și $x \leq 1$.

288. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii lui x la y și cu $[x]$ partea întreagă a numărului real x . Care este valoarea afișată dacă se introduc de la tastatură valorile 59, 480, 16, 329, 0?

- A. 3145;
- B. 5413;
- C. 2689.

```
citeste n {n nr natural}
a←0; p←1
┌cat timp n≠0 exec
│ ┌cat timp n>9 execută
│ │ n←[n/10]
│ └─┘
│ a←n*p+a ; p←p*10
│ citeste n
└─┘
scrie a
```

289. Condiția ca două numere întregi a și b să fie ambele nenule este:

- A. $(a \neq 0)$ sau $(b \neq 0)$;
- B. $a * b \neq 0$;
- C. $a + b \neq 0$.

290. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii lui x la y și cu $[x]$ partea întreagă a numărului real x . Ce se va afișa pentru $a=204$ și $b=212$?

- A. 3;
- B. 4;
- C. 5.

```

citește a,b
(a,b nr. naturale, a<=b)
k←0
pentru i←a,b,1 execută
  n←i; c←0
  cat timp n>0 execută
    dacă n%10%2=1 atunci
      c←1
    n←[n/10]
  dacă c=0 atunci
    k←k+1
scrie k

```

291. Care dintre variante inserează o cifră c în fața ultimei cifre a unui număr natural n ?

- A. $n=(n\%10*10+c)*10+n/10$;
- B. $n=n/10+c+n\%10$;
- C. $n=(n/10*10+c)*10+n\%10$.

292. Care dintre următoarele secvențe calculează suma elementelor de pe linia k a unei matrice a , cu m linii (numerotate de la 1 la m) și n coloane (numerotate de la 1 la n)?

- A. $s=0$; for($i=m$; $i>0$; $i--$) $s=s+a[k][i]$;
- B. $s=0$; $i=1$; while($i<=n$) { $s=s+a[k][i]$; $i++$;};
- C. $s=0$; for($i=n$; $i>=1$; $i--$) $s=s+a[i][k]$;

293. Se consideră programul pseudocod alăturat. Ce va afișa algoritmul pentru $a=3$ și $b=10$?

- A. 6;
- B. 8;
- C. 4.

```

citește a,b {a,b ∈ N}
dacă a<b atunci
  a←a-b
  b←a+b
  a←b-a
k←0
cât timp a≥b execută
  a←a-b
  k←k+2
scrie k

```

294. Care dintre următoarele atribuiri elimină cifra din mijloc a unui număr natural n cu exact 5 cifre?

- A. $n=n\%1000*100+n/100$;
- B. $n=n/1000*100+n\%100$;
- C. $n=n/1000+n\%100$.

295. Un program care urmează să sorteze crescător 15 numere reale aflate într-un fișier, citește aceste numere într-o variabilă x. Cum trebuie declarată această variabilă?

- A. int x;
- B. float x[15];
- C. unsigned x[15];

296. Care este valoarea tipărită de secvența alăturată

- A. 3;
- B. 3.57;
- C. 3.5714.

```
int a=25,b=7; float f;
f=(float)a/b;
f=(int)(f*100);
f=f/100;
cout<<f;
```

297. Câte atribuiri se execută în secvența alăturată, pentru n=245?

- A. 5;
- B. 7;
- C. 3.

```
s=0;
while (n!=0) {
    s=s+1;
    n=n/100;
}
```

298. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii numerelor întregi x și y. Ce va tipări algoritmul pentru a=2 și b=11?

- A. 5;
- B. 7;
- C. 9.

```
citește a,b {a,b ∈ N}
dacă a%2=0 atunci
    a ← a+1
s ← 0
cât timp a ≤ b execută
    a ← a+2
    s ← s+1
scrie s
```

299. Care dintre variante nu reprezintă o declarație corectă a unei variabile de tip caracter?

- A. char ab;
- B. char a&b;
- C. char a_b;

300. O secvență de instrucțiuni echivalentă cu secvența alăturată care să conțină o singură instrucțiune if este:

- A. if(x>y&& y>z) s=x+y+z; else p=x*y*z;
- B. if(x>y&& y>z) p=x*y*z;
- C. if(x>y|| y>z) s=x+y+z;

```
if(x>y)
if(y>z)
    if(z>x)s=x+y+z;
else p=x*y*z;
```

301. Știind că valoarea inițială a variabilei k este un număr natural par cu cel mult 4 cifre, stabiliți valoarea tipărită de secvența alăturată.

- A. 1;
- B. 5;
- C. 0.

```
while (k>1)
k=k-2;
n=abs(k-5);
cout<<n;
```

302. Care este valoarea inițială a variabilei n astfel încât, la sfârșitul executării secvenței alăturate, variabila întreagă c să aibă valoarea 3?

- A. 123;
- B. 10020;
- C. 5000.

```
c=0;
while (n%10==0){
    n=n/10;
    c++;
}
```

303. În programul pseudocod alăturat. s-a notat cu $x\%y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x. Ce va tipări algoritmul pentru 2793?

- A. 32;
- B. 64;
- C. 16.

```
citește n {n ∈ N}
a ← n%10 ; m ← a
cât timp n > 9 execută
    n ← [n/10]; b ← n%10
    dacă a > b atunci
        m ← m*10 + b ; a ← b
scrie m
```

304. Care dintre secvențele de mai jos conduce la memorarea în variabila max a celei mai mari dintre cele n valori întregi din tabloul unidimensional a (în care a[0] este primul element) ?

- A. for(i=0;i<n-1;i++) if(a[i]>a[i+1])max=a[i];
- B. for(i=0;i<n-1;i++)
 - if(a[i]>a[i+1]){max=a[i];a[i]=a[i+1];a[i+1]=max;}
 - max=a[n-1];
- C. max=a[0];
 - for(i=1;i<n;i++) if(a[i]<max) max=a[i];

305. Care dintre următoarele expresii este echivalentă cu expresia alăturată ?

- A. $a \leq b \parallel a \geq c \ \&\& \ a \leq d$
- B. $(a > b \parallel a < c) \ \&\& \ a > d$
- C. $a > b \ \&\& \ a < c \parallel a > d$

```
!((a<=b && a>=c) || a<=d)
```

306. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural y, iar cu $[x]$ partea întreagă a numărului real x. Ce se afișează dacă numărul citit este 6?

- A. 2;
- B. 4;
- C. 6.

```
citește a (a nr natural, a > 1)
n ← 0 ; d ← 2
cât timp d ≤ a execută
    i ← 0
    cât timp a % d = 0 atunci
        a ← [a/d]
        i ← i + 1
    n ← n + i ; d ← d + 1
scrie n
```

307. Variabila întregă a reține un număr natural format din exact două cifre. Care dintre următoarele instrucțiuni atribuie variabilei întregi b o valoare egală cu suma cifrelor numărului memorat în variabila a?

- A. $b = a \% 100;$
- B. $b = a \% 10 + a / 10;$
- C. $b = a \% 10 + a / 100;$

308. Se consideră un tablou unidimensional a în care $a[0]=4$, $a[1]=2$, $a[2]=5$, $a[3]=1$, $a[4]=3$. Care va fi conținutul său după executarea secvenței alăturate (în care variabilele i și aux sunt de tip int)?

- A. $a[0]=2$, $a[1]=4$, $a[2]=1$, $a[3]=3$, $a[4]=5;$
- B. $a[0]=4$, $a[1]=2$, $a[2]=5$, $a[3]=1$, $a[4]=3;$
- C. $a[0]=1$, $a[1]=2$, $a[2]=3$, $a[3]=4$, $a[4]=5.$

```
for (i=0;i<4;i++)
  if (a[i]>a[i+1]){
    aux=a[i];a[i]=a[i+1];a[i+1]=aux;
  }
```

309. Se consideră programul pseudocod alăturat. S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural y, iar cu $[x]$ partea întregă a numărului x. Ce se afișează dacă numerele citite sunt 24 și 36?

- A. 2;
- B. 3;
- C. 4.

```
citește a,b;
(numere naturale nenule)
c←0
repetă
  i←a%2
  j←b%2
  dacă i+j=0 atunci
    c←c+1
  a←a*i+(1-i)*[a/2]
  b←b*j+(1-j)*[b/2]
  până când i*j=1
scrie c
```

310. Se consideră programul pseudocod alăturat. S-au folosit următoarele notații: mod pentru restul împărțirii întregi și div pentru câtul împărțirii întregi. Indicați cea mai mare valoare posibilă pentru x astfel încât algoritmul să afișeze valoarea 2355.

- A. 2399;
- B. 2389;
- C. 2379.

```
citește x {nr.natural}
v←0; z←1;
repetă
  a←x mod 10
  dacă a>5 atunci
    v←v+z*5
  altfel
    v←v+z*a
  x←x div 10
  z←z*10
  până când x=0
scrie v
```


311. Se consideră un tablou unidimensional a în care elementele sunt, în ordine : 1,3,5,7,10,16,21. Pentru a afla poziția pe care se află valoarea $x=10$ se aplică metoda căutării binare. Care este succesiunea corectă de elemente a căror valoare se compară cu valoarea lui x ?

- A. 21,16,10 ;
- B. 7,16,10 ;
- C. 1,3,5,7,10.

312. Se consideră programul pseudocod alăturat. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural y , iar cu $[x]$ partea întreagă a numărului real x . Ce afișează algoritmul dacă se citește valoarea 15793?

- A. 26904;
- B. 26902;
- C. 26804.

```
citește n (număr natural, nenul)
p ← 1
┌cât timp p<n execută
│ n ← ([n/p]+1)*p+n%p
│ p ← p*10
│ ■
└scrie n
```

313. Care dintre expresiile următoare, scrise în limbajul C/C++, are valoarea 1?

- A. 'a'=='A';
- B. '1'+ '2'=='3';
- C. 'a'<'b'.

314. Ce valoare are variabila întreagă n în urma executării secvenței alăturate?

- A. 10;
- B. 12;
- C. 11.

```
n=0;
do
  n=n+3;
while(n<=10);
```

315. Fie tabloul unidimensional a în care elementele sunt, în ordine : 1,3,5,7,10,16,21. Pentru a verifica dacă numărul $x=4$ se află printre elementele tabloului, se aplică metoda căutării binare. Care este succesiunea corectă de elemente cu care se compară x ?

- A. 1,3,5 ;
- B. 7,5,3 ;
- C. 7,3,5.

316. Se consideră două tablouri unidimensionale A și B: $A=(1,3,5,9,10)$ respectiv $B=(2,4,6,7)$. În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:

- A. (1,2,3,4,5,6,9,7,10) ;
- B. (1,2,3,4,5,6,7,9,10) ;
- C. nu se poate realiza interclasarea.

317. Ce valoare are variabila întreagă n în urma executării secvenței alăturate?

- A. 11;
- B. 12;
- C. 10.

```
n=0;
while(n<=11) n=n+2;
```

318. Știind că variabila reală x are o valoare ce aparține intervalului închis $[3,7]$, care dintre expresiile de mai jos, scrise în limbajul C/C++, NU are valoarea 1?

- A. $!(x < 3 \parallel x > 7)$;
- B. $3 < x \ \&\& \ x \leq 7$;
- C. $x \geq 3 \ \&\& \ x \leq 7$.

319. Se consideră programul pseudocod alăturat. S-a notat cu a_k elementul aflat pe poziția k în tabloul unidimensional cu numele a , cu $|x|$ valoarea absolută a numărului natural x și cu $[x]$ partea întreagă a numărului x . Ce se afișează dacă sunt citite, în ordine, numerele 3, 5, 8, 1, 7?

- A. 1;
- B. 3;
- C. 5.

```

┌ pentru i ← 1,5 execută
│ citește ai (număr întreg)
│ ■
└ pentru i ← 1,4 execută
  ┌ m ← ai + ai+1 + |ai - ai+1|
  │ ai+1 ← ai + ai+1 - [m/2]
  │ ai ← [m/2]
  │ ■
  └ scrie a5

```

320. Algoritmul următor testează dacă șirul de caractere s este palindrom (citit atât de la stânga la dreapta, cât și de la dreapta la stânga, se obține același șir de caractere). Care este expresia care poate înlocui punctele de suspensie astfel încât algoritmul să fie corect?

- A. $i == j$;
- B. $i >= j$;
- C. $i < j$.

```

i = 0; j = strlen(s) - 1;
while(i < j && s[i] == s[j]){
    i = i + 1; j = j - 1;
}
if(.....)
    cout << "palindrom";
else
    cout << "nu e palindrom";

```

321. Vectorul A conține, începând cu indicele 1 elementele (1,2,2,3,3,3,4,4,4,4,...) Care este valoarea elementului cu indicele 25 ?

- A. 25;
- B. 7;
- C. 6.

322. Se consideră programul pseudocod alăturat în care s-a notat cu $x \% y$ restul împărțirii întregi a lui x la y și cu $[x]$ partea întreagă a numărului real x . Ce se va afișa pentru $n = 45$?

- A. 5;
- B. 4;
- C. 6.

```

citește n (nr. natural, n > 1)
d ← 2 (d număr natural)
┌ cat timp n % d ≠ 0 execută
│ d ← d + 1
│ ■
└ cat timp n % d = 0 execută
  ┌ n ← [n/d]
  │ ■
  └ dacă n = 1 atunci
    ┌ scrie d
    │ altfel
    └ scrie n
    └ ■

```


328. Pentru a verifica dacă toate elementele unui vector A sunt strict mai mici decât toate elementele unui alt vector B ordonat crescător, se compară pe rând fiecare element din A cu toate elementele din B. Această metodă de verificare este:
- A. incorectă;
 - B. corectă dar neeficientă;
 - C. corectă numai pentru componente întregi.

329. Se consideră programul pseudocod alăturat. Ce se va afișa pentru $x=1, y=10$?
- A. 10;
 - B. 1;
 - C. 0.

```

citește x,y (numere naturale , x<y)
k←0
┌cat timp x<y execută
│ x←x+1 ; y←y-1
│ k←k+1
│ ──┘
│
│ ┌dacă x=y atunci
│ │ scrie 2*k+1
│ └altfel
│   scrie 2*k
└─┘

```

330. Se consideră programul pseudocod alăturat. Câte perechi (x,y) există în intervalul [1;10] astfel încât să se afișeze valoarea 5?
- A. 6;
 - B. 5;
 - C. 1.

```

citește x,y (numere naturale , x<y)
k←0
┌cat timp x<y execută
│ x←x+1 ; y←y-1
│ k←k+1
│ ──┘
│
│ ┌dacă x=y atunci
│ │ scrie 2*k+1
│ └altfel
│   scrie 2*k
└─┘

```

331. Condiția ca numărul natural x să fie multiplu al numerelor naturale a și b este:
- A. $a \% x == 0 \parallel b \% x == 0$;
 - B. $x \% a == 0 \ \&\& \ x \% b == 0$;
 - C. $x \% a == 0 \parallel x \% b == 0$.
332. Niciunul dintre numerele reale x și y nu aparține intervalului [a,b] dacă și numai dacă:
- A. $(x < a \parallel x > b) \ \&\& \ (y < a \parallel y > b)$;
 - B. $x < a \ \&\& \ y < a \parallel x > b \ \&\& \ y > b$;
 - C. $x < a \parallel x > b \ \&\& \ y < a \parallel y > b$.
333. Pentru a verifica dacă toate elementele unui vector ordonat descrescător A sunt strict mai mici decât toate elementele unui alt vector B ordonat crescător, se compară primul element din A cu primul element din B. Această metodă de verificare este:
- A. corectă numai pentru componente întregi;
 - B. corectă și neeficientă;
 - C. corectă și eficientă.

334. Se consideră programul pseudocod alăturat în care s-a folosit notația $[a]$ pentru partea întreagă a numărului real a . Care este valoarea afișată pentru $n=1234$?

- A. 0;
- B. 1;
- C. 4.

```

citește n (număr întreg, n>0)
k ← 0 ; c ← 0 ; naux ← n
┌ cat timp naux > 0 execută
│   naux ← [naux/10]
│   k ← k+1
│   c ← c*10+1
│   ──┘
└─┘
┌ pentru i ← 1, k execută
│   n ← n-c
│   c ← [c/10]
│   ──┘
└─┘
scrie n

```

335. Pentru căutarea unei valori între elementele unui vector ordonat descrescător, vom utiliza un algoritm eficient de tip:

- A. interschimbare;
- B. quick sort;
- C. căutare binară.

336. Într-o matrice pătratică de dimensiune n , notăm cu $a(i,j)$ elementul situat pe linia i și coloana j ($1 \leq i \leq n$ și $1 \leq j \leq n$). Diagonala principală și cea secundară determină în matrice patru zone triunghiulare. Ce condiție trebuie să îndeplinească indicii elementelor din triunghiul superior al matricei?

- A. $i < j$ și $(i+j) < n+1$;
- B. $i < j$ sau $(i+j) < n$;
- C. $i > j$ și $(i+j) < n$.

337. Secvența alăturată de program, va afișa:

- A. toate numerele naturale de două cifre;
- B. numerele naturale impare mai mici decât 100;
- C. toate numerele întregi mai mici decât 99.

```

a=99;
while (a>=1)
{ cout<<a;
a=a-2; }

```

338. Secvența de program alăturată va afișa:

- A. numerele naturale din intervalul $[1,27]$;
- B. numerele naturale din intervalul $[65,90]$;
- C. literele mari ale alfabetului englez.

```

char c;
for(c='A';c<='Z';c++)
    cout<<c;

```

339. Se consideră secvența de program alăturată. Instrucțiunea de afișare se execută de:

- A. 100 ori;
- B. 45 ori;
- C. 20 ori.

```

for(i=1;i<=10;i++)
for(j=i+1;j<=10;j++)
cout<<j;

```

340. Se consideră programul pseudocod alăturat. Care sunt valorile afișate pentru $a=7$?

- A. 6 și 12;
- B. 6 și 11;
- C. 7 și 12.

```

citește a (număr natural, a>0)
b ← (a+2)*(a+3) ; k ← 0
cât timp (b-a ≥ 0) execută
    b ← b-a
    k ← k+1
scrie b, k
    
```

341. Secvența alăturată afișează:

- A. numere naturale cu cifre distincte, mai mici decât S;
- B. numerele naturale de două cifre care au suma cifrelor egală cu S;
- C. numerele naturale care au suma cifrelor egală cu S.

```

citește S (numar natural, S ≤ 18)
pentru i=1,9 execută
    dacă (S-i) ≥ 0 și (S-i) ≤ 9 atunci
        afișează 10*i+(S-i)
    
```

342. Într-o matrice cu 10 linii și 20 de coloane, dorim să inserăm o nouă coloană având toate elementele 0 după cea de-a treia coloană a acestei matrice. Pentru a realiza această operație:

- A. vom deplasa toate coloanele, începând de la a patra până la ultima, cu o poziție spre dreapta și pe coloana a patra, pe toate liniile vom memora 0;
- B. vom deplasa toate coloanele, începând de la a treia până la prima, cu o poziție spre stânga și pe coloana a patra, pe toate liniile vom memora 0;
- C. vom deplasa toate coloanele, începând de la ultima până la a patra, cu o poziție spre dreapta și pe coloana a patra, pe toate liniile vom memora 0.

343. Se consideră programul pseudocod alăturat. S-a notat cu $[x]$ partea întreagă a numărului real x , și cu $x\%y$ restul împărțirii lui x la y . Care este valoarea afișată pentru $a=19$ și $b=45$?

- A. 855;
- B. 655;
- C. 655.

```

citește a, b (numere naturale)
s ← 0
repetă
    dacă a%2 ≠ 0 atunci
        s ← s+b
    a ← [a/2] ; b ← b*2
până când a < 1
scrie s
    
```

344. În urma executării secvenței alăturate de program, se va construi matricea:

- | | | |
|----------|----------|--------------|
| A. 2 3 4 | B. 2 4 5 | C. 2 3 4 5 6 |
| 3 4 5 | 3 4 5 | 3 4 5 6 7 |
| 4 5 6 | 4 5 6 | 4 5 6 7 8 |
| 5 6 7 | | |
| 6 7 8 | | |

```

for(j=1; j ≤ 5; j++)
    for(i=1; i ≤ 3; i++)
        a[i][j]=i+j;
    
```

345. Se consideră secvența de instrucțiuni în pseudocod alăturată. Ce valoare trebuie scrisă în pătrățel pentru a se afișa 165?

- A. 8;
- B. 9;
- C. 10.

```

i ← 3; E ← 1
cât timp i < □ execută
    dacă i % 2 ≠ 0 atunci
        E ← E + i * i
    i ← i + 1
scrie E
    
```

346. Se considera tabloul unidimensional a=(1,2,3,4,5). Care va fi rezultatul afișat în urma executării secvenței alăturate?

- A. 1151515;
- B. 1361015;
- C. 6101315.

```

for (i=0; i<5; i++) {
    s=0;
    for(j=0; j<=i; j++)
        s=s+a[j];
    cout<<s;
}
    
```

347. Se consideră programul pseudocod alăturat. S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x . Ce valoare se va afișa pentru $n=480$?

- A. 7;
- B. 8;
- C. 4.

```

citește n(număr natural)
s ← 0; f ← 2
cât timp n > 1 execută
    p ← 0
    cât timp n % f = 0 execută
        n ← [n/f]; p ← p + 1
    dacă p ≠ 0 atunci
        s ← s + p
    f ← f + 1
scrie s
    
```

348. Se consideră programul pseudocod alăturat. S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x . Stabiliți câte valori din intervalul $[10,40]$ pot fi introduse pentru variabila n , astfel încât după executarea programului, valoarea afișată să fie 1.

- A. 8;
- B. 38;
- C. 28.

```

citește n(număr natural)
s ← 0; f ← 2
cât timp n > 1 execută
    p ← 0
    cât timp n % f = 0 execută
        n ← [n/f]; p ← p + 1
    dacă p ≠ 0 atunci
        s ← s + p
    f ← f + 1
scrie s
    
```

349. Stabiliți care dintre următoarele expresii este adevărată dacă și numai dacă numărul întreg x , nu aparține intervalului închis $[20,100]$.

- A. $(x \leq 20) \vee (x \geq 100)$;
- B. $(x < 20) \vee \neg(x > 100)$;
- C. $(x \leq 19) \vee (x \geq 101)$.

350. Fie vectorii a și b în care $a = (1,2,4,3,5)$ și $b = (4,2,1,3)$. Atunci $a[b[0]]$ are valoarea:

- A. 5;
- B. 3;
- C. 2.

351. Știind că două intervale de numere reale $[a,b]$ și $[c,d]$ îndeplinesc condiția ca maximul dintre a și c este mai mic sau egal decât minimul dintre b și d , atunci intervalul $[\max\{a,c\}, \min\{b,d\}]$ reprezintă:

- A. $[a,b] \cap [c,d]$;
- B. $[a,b] \cup [c,d]$;
- C. $[c,d] - [a,b]$.

352. Se consideră programul pseudocod alăturat, unde x div y înseamnă câtul împărțirii numerelor întregi x și y . Care este valoarea afișată pentru $n=81832$?

- A. 8;
- B. 1;
- C. 3.

```
citește n (n ∈ N, n > 9)
p ← 10 ; q ← 1
┌repetă
│  p ← p * 10
│  q ← q * 10
└până când q ≤ n și n ≤ p
scrie n div q
```


Răspunsuri:

1-C	33-B	65-A	97-C	129-C	161-C	193-B	225-C	257-C	289-B	321-B
2-C	34-A	66-C	98-C	130-C	162-B	194-C	226-B	258-A	290-A	322-A
3-A	35-B	67-B	99-B	131-C	163-C	195-A	227-C	259-A	291-C	323-B
4-A	36-A	68-B	100-C	132-B	164-B	196-C	228-A	260-B	292-B	324-A
5-C	37-C	69-A	101-C	133-C	165-A	197-C	229-B	261-B	293-A	325-A
6-C	38-C	70-B	102-C	134-B	166-C	198-B	230-A	262-B	294-B	326-A
7-A	39-A	71-A	103-C	135-B	167-B	199-B	231-A	263-A	295-B	327-B
8-B	40-C	72-B	104-C	136-C	168-B	200-C	232-B	264-A	296-B	328-B
9-A	41-C	73-C	105-C	137-A	169-C	201-C	233-B	265-C	297-A	329-A
10-B	42-B	74-C	106-C	138-C	170-C	202-C	234-C	266-C	298-A	330-A
11-B	43-C	75-A	107-C	139-C	171-A	203-C	235-B	267-B	299-B	331-B
12-A	44-C	76-B	108-C	140-C	172-B	204-A	236-A	268-B	300-B	332-A
13-A	45-A	77-A	109-C	141-B	173-A	205-C	237-A	269-C	301-B	333-C
14-C	46-C	78-C	110-B	142-C	174-C	206-B	238-A	270-A	302-C	334-A
15-C	47-A	79-C	111-C	143-C	175-B	207-A	239-C	271-C	303-A	335-C
16-B	48-C	80-C	112-C	144-A	176-A	208-B	240-B	272-B	304-B	336-A
17-A	49-C	81-C	113-B	145-C	177-A	209-A	241-A	273-A	305-B	337-B
18-C	50-B	82-C	114-C	146-C	178-B	210-B	242-B	274-B	306-A	338-C
19-A	51-B	83-C	115-A	147-A	179-C	211-A	243-B	275-A	307-B	339-B
20-B	52-B	84-C	116-C	148-C	180-C	212-B	244-A	276-C	308-A	340-A
21-C	53-C	85-A	117-C	149-B	181-C	213-A	245-C	277-C	309-A	341-B
22-B	54-A	86-C	118-B	150-C	182-B	214-A	246-A	278-A	310-A	342-C
23-C	55-B	87-C	119-C	151-C	183-B	215-B	247-B	279-A	311-B	343-A
24-B	56-A	88-C	120-C	152-C	184-B	216-C	248-A	280-B	312-A	344-C
25-B	57-A	89-C	121-C	153-B	185-C	217-B	249-C	281-A	313-C	345-C
26-A	58-C	90-B	122-B	154-C	186-C	218-C	250-B	282-B	314-B	346-B
27-A	59-C	91-B	123-B	155-C	187-B	219-B	251-B	283-C	315-C	347-A
28-A	60-C	92-C	124-C	156-C	188-C	220-A	252-B	284-C	316-B	348-A
29-C	61-C	93-C	125-C	157-A	189-A	221-C	253-C	285-A	317-B	349-C
30-C	62-A	94-C	126-C	158-C	190-C	222-C	254-C	286-A	318-B	350-A
31-B	63-B	95-B	127-C	159-A	191-C	223-B	255-A	287-B	319-A	351-A
32-A	64-C	96-C	128-B	160-C	192-B	224-B	256-C	288-A	320-B	352-A