

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**MODULUL 1**

***Limbaje și tehnici de programare***

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Programare procedurala**

## PROGRAMARE PROCEDURALA

1. În care dintre variantele de mai jos se declară un tablou unidimensional (vector)  $x$  în care se pot memora cel mult 100 de numere reale?

- a) `x=float[100];`
- b) `double x[100];`
- c) `float x[100];`
- d) `real x(100);`

2. Care dintre următoarele expresii logice este adevărată (are o valoare nenulă) dacă și numai dacă numărul real memorat în variabila  $x$  nu aparține intervalului  $(0,5]$ ?

- a) `(x<=0) || (x>5)`
- b) `(x<=0) && (x>5)`
- c) `(x<0) || (x>=5)`
- d) `(x<=0) && (x>=5)`

3. Care dintre următoarele expresii este adevărată (are o valoare nenulă) dacă și numai dacă numărul întreg memorat în variabila  $x$  aparține intervalului  $(1,6]$ ?

- a) `(x>=1) || (x<6)`
- b) `(x>1) || (x<=6)`
- c) `(x>1) && (x<6)`
- d) `(x>1) && (x<=6)`

4. După executarea instrucțiunii `float x = 27/5*2/3*7;` ce valoare va fi memorată în variabila  $x$ ?

- a) 25.2
- b) 21.0
- c) 6.3
- d) 7.0

5. După executarea instrucțiunii `float x = 55/17*5/8+48/5/8*15;` ce valoare va fi memorată în variabila  $x$ ?

- a) 17.022058
- b) 17.0
- c) 16.0
- d) 21.219914

6. Se consideră următoarea secvență de instrucțiuni:

```
int t=0,a=1234,b=10;
while (a>=b)
{
    a=a-b;
    t++;
}
```

```
}
printf("%d %d", t, a);
```

Ce valori vor fi afișate pe ecran după executarea secvenței de mai sus?

- a) 124 4
- b) 123 4**
- c) 123 5
- d) 124 3

7. Care dintre următoarele secvențe de instrucțiuni afișează valoarea 654, știind că *s* și *i* sunt două variabile de tip întreg?

- a) `s=0;`  
`for(i=0;i<=654;i++) s++;`  
`printf("%d",s);`
- b) `s=651;`  
`while(s<=654) s++;`  
`printf("%d",s);`
- c) `for(i=1;i<=3;i++) printf("%d",7-i);`**
- d) `s=7;`  
`while(s>=1) printf("%d",s-1);`

8. Care dintre următoarele secvențe de instrucțiuni afișează valoarea 5432, știind că *s* și *i* sunt două variabile de tip întreg?

- a) `s=0;`  
`for(i=0;i<=5432;i++) s++;`  
`printf("%d",s);`
- b) `s=5421;`  
`while(s<=5432) s++;`  
`printf("%d",s);`
- c) `for(i=1;i<4;i++) printf("%d",6-i);`
- d) `s=6;`  
`while(s>=3) printf("%d",--s);`**

9. Considerăm următoarea secvență de instrucțiuni:

```
int s=0;
while(n>0)
{
    if(n%10>s) s=n%10;
    else s=10;
```

```

    n=n/10;
}
printf("%d",s);

```

Știind că variabilele  $s$  și  $n$  sunt de tip întreg, ce valoare se va afișa după executarea secvenței de mai sus pentru  $n = 9321$ ?

- a) 9
- b) 10
- c) 15
- d) 1

10. Care dintre următoarele secvențe de instrucțiuni afișează câtul și restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$ ?

a) 

```
int t=0;
while(a>=b)
{
    a=a-b;
    t++;
}
printf("%d %d",t,a);
```

b) 

```
int t=0;
do
{
    a=a-b;
    t++;
}while(a>=b);
printf("%d %d",t,a);
```

c) 

```
int t=0;
while(a!=b)
{
    a=a-b;
    t++;
}
printf("%d %d",t,b);
```

d) 

```
int t=0;
while(a%b==0)
{
    a=a-b;
    t++;
}
printf("%d %d",t,a);
```

11. Considerăm următorul program:

```
#include <stdio.h>
```

## PROGRAMARE PROCEDURALA

```
void sch(int a, int *b)
{
    int aux;
    aux = a;
    a = *b;
    *b = aux;
}

int main()
{
    int x = 1, y = 2;
    sch(x, &y);
    printf("%d", x+y);
    return 0;
}
```

Ce valoare se va afișa pe ecran după executarea programului de mai sus?

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

12. Considerăm următorul program:

```
#include <stdio.h>
void sch(char a, char *b)
{
    char aux;
    aux = a;
    a = *b;
    *b = aux;
}

int main()
{
    char x = '1', y = '2';
    sch(x, &y);
    printf("%c,%c", x, y);
    return 0;
}
```

Ce valori se vor afișa pe ecran după executarea programului de mai sus?

- a) 1,2
- b) 2,1
- c) 1,1
- d) 2,2

13. Considerăm următorul program:

## PROGRAMARE PROCEDURALA

```
#include <stdio.h>
void sch(int *a, int b)
{
    int aux;
    aux = *a;
    *a = b;
    b = aux;
}

int main()
{
    int x = 1, y = 2;
    sch(&x, y);
    printf("%d", x*y);
    return 0;
}
```

Ce valoare se va afișa pe ecran după executarea programului de mai sus?

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

14. Considerăm următorul program:

```
#include<stdio.h>
void f(int a,int *b)
{
    a++;
    *b=a;
    (*b)++;
}

void g(int *a,int b)
{
    b++;
    *a=b;
    (*a)++;
}

int main()
{
    int x=4, y=-2;
    f(x,&y);
    g(&x,y);
    printf("%d %d",x,y);
    return 0;
}
```

Ce valori se vor afișa pe ecran după executarea programului de mai sus?

- a) 4 8
- b) 8 8
- c) 8 6
- d) 6 6

15. Care dintre următoarele secvențe de instrucțiuni atribuie variabilei de tip întreg *max* cea mai mare valoare din tabloul *a*, format din *n* numere întregi?

- a) 

```
max=0;
for(i=0;i<n;i++)
    if(a[i]>max) max=a[i];
```
- b) 

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

```
max=a[0];
for(i=0;i<n;i++)
    if(a[i]>max) max=a[i];
```
- d) 

```
max=0;
for(i=0;i<n-1;i++)
    if(a[i]<a[i+1]) max=a[i+1];
```

16. Considerăm următoarea secvență de instrucțiuni:

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

Cu ce expresie dintre cele de mai jos trebuie înlocuite spațiile punctate din secvență de instrucțiuni dată astfel încât aceasta să afișeze câte valori strict pozitive și pare sunt în tabloul *a*, format din *n* numere întregi?

- a) `(a[i]>0) && (a[i]%2!=0)`
- b) `(a[i]>0) && (a[i]%2==0)`
- c) `(a[i]>0) || (a[i]%2!=0)`
- d) `(a[i]>=0) || (a[i]%2==0)`

17. Care este valoarea expresiei `strlen("programare")+strcmp("test","test")`?

- a) 10
- b) 14
- c) 18
- d) "programaretesttest"

18. Considerăm următoarea secvență de instrucțiuni:



```
char s[100];
strcpy(s, "");
strcat(s, "abcdefgh");
strcpy(s+2, s+4);
printf("%s %d", s, strlen(s));
```

Ce se va afișa pe ecran după executarea secvenței date?

- a) adefgh 6
- b) abefgh 6**
- c) abfgh 5
- d) abefgh 8

19. Care din următoarele expresii de tip logic este adevărată (are o valoare nenulă) dacă și numai dacă șirul de caractere *s*, de lungime 10, este obținut prin concatenarea a două șiruri identice?

- a) strcmp(s, s+5) == 0
- b) s == strstr(s, s+5)**
- c) s == s+5
- d) strcmp(s, strcat(s, s+5)) == 0

20. Considerăm următoarea secvență de instrucțiuni:

```
char s[] = "abcdabcd", c = 'c';
char *p = strchr(s, c);
printf("%d", p - s);
```

Ce se va afișa pe ecran după executarea secvenței date?

- a) cdabcd
- b) 6
- c) cd
- d) 2**

21. Considerăm următoarea secvență de instrucțiuni:

```
char s[20];
strcpy(s, "abcdabcd");
strncat(s, s+2, 3);
strcpy(s, s+4);
printf("%d", strlen(s));
```

Ce se va afișa pe ecran după executarea secvenței date?

- a) 6
- b) 10

- c) 9
- d) 7

22. Considerăm următoarea secvență de instrucțiuni:

```
char s[20];
strncpy(s, "abcdabcd", 6);
s[6]='\0';
strcat(s, s+4);
strcpy(s+3, s+6);
printf("%s", s);
```

Ce se va afișa pe ecran după executarea secvenței date?

- a) abcabab
- b) abcdab
- c) **abcab**
- d) abcdabd

23. Considerăm următoarele structuri:

```
typedef struct
{
    int zi, luna, an;
}Data;

typedef struct
{
    char nume[30];
    Data data_nasterii;
    float media;
}Student;
```

Știind că variabila *st* este de tip *Student*, indicați instrucțiunea de mai jos prin care luna nașterii studentului respectiv primește valoarea 12:

- a) *st->data\_nasterii->luna=12;*
- b) ***st.data\_nasterii.luna=12;***
- c) *data\_nasterii.luna=12;*
- d) *st.luna=12;*

24. Considerăm următoarele structuri:

```
typedef struct
{
    int zi, luna, an;
}Data;

typedef struct
```

```
{
    char nume[30];
    Data data_nasterii;
    float media;
}Student;
```

Știind că variabila *st* este de tip *Student*, indicați instrucțiunea de mai jos prin care anul nașterii studentului respectiv primește valoarea 1990:

- a) *st->data\_nasterii->an=1990;*
- b) *st.data\_nasterii.an=1990;***
- c) *data\_nasterii.an=1990;*
- d) *st.an=1990;*

25. Considerăm următoarele structuri:

```
typedef struct
{
    double x,y;
}Punct_2D;
```

```
typedef struct
{
    Punct_2D p;
    double z;
}Punct_3D;
```

Știind că variabila *a* este de tip *Punct\_3D*, fiind folosită pentru a stoca coordonatele unui punct în spațiu, indicați instrucțiunea de mai jos prin care toate cele 3 coordonate ale punctului *a* se inițializează cu valoarea 0:

- a) *a.p.x = a.p.y = a.p.z = 0.0;*
- b) *a.p.x = a.p.y = a.z = 0.0;***
- c) *a.x = a.y = a.z = 0.0;*
- d) *a.p = a.z = 0.0;*

26. Considerăm tipul de date *Punct*, capabil să memoreze abscisa și ordonata unui punct din plan, și tipul de date *Segment*, capabil să memoreze două puncte reprezentând extremitățile unui segment din plan, definite astfel:

```
typedef struct
{
    double x,y;
}Punct;
```

```
typedef struct
{
    Punct A,B;
}Segment;
```

Care dintre următoarele expresii are o valoare nenulă dacă și numai dacă variabila  $s$  de tip `Segment` memorează informații despre un segment vertical (aflat pe axa  $Oy$  sau paralel cu axa  $Oy$ )?

- a) `s.A == s.B`
- b) `s.x == s.y`
- c) `A.x == B.x`
- d) `s.A.x == s.B.x`

27. Considerăm tipul de date `Punct`, capabil să memoreze abscisa și ordonata unui punct din plan, și tipul de date `Segment`, capabil să memoreze două puncte reprezentând extremitățile unui segment din plan, definite astfel:

```
typedef struct
{
    float x,y;
}Punct;
```

```
typedef struct
{
    Punct A,B;
}Segment;
```

Care dintre următoarele funcții returnează lungimea segmentului transmis prin intermediul parametrului  $s$  de tip `Segment`?

- a) 

```
double f(Segment s)
{
    return pow(s.A.x-s.B.x,2)+pow(s.A.y-s.B.y,2);
}
```
- b) 

```
double f(Segment s)
{
    return sqrt((s.A.x-s.B.x)+(s.A.y-s.B.y));
}
```
- c) 

```
double f(Segment s)
{
    return s.B-s.A;
}
```
- d) 

```
double f(Segment s)
{
    return sqrt(pow(s.A.x-s.B.x,2)+pow(s.A.y-s.B.y,2));
}
```

28. Considerăm funcția `int suma(int x,int y)` care returnează suma numerelor întregi  $x$  și  $y$ , precum și funcția `int prod(int x,int y)` care returnează produsul numerelor întregi

## PROGRAMARE PROCEDURALA

$x$  și  $y$ . Știind că  $a$ ,  $b$  și  $c$  sunt 3 variabile de tip întreg, care dintre expresiile de mai jos atribuie variabilei  $t$  de tip întreg valoarea expresiei  $(a+b) * (a+c) + b*c$ ?

- a) `t = prod(suma(a,b), suma(a,c), prod(b,c));`
- b) `t = suma(prod(suma(a,b), suma(a,c)), suma(b,c));`
- c) `t = prod(suma(a,b), suma(a,c)+suma(b,c));`
- d) `t = suma(prod(suma(a,b), suma(a,c)), prod(b,c));`

29. Considerăm funcția `int suma(int x, int y)` care returnează suma numerelor întregi  $x$  și  $y$ , precum și funcția `int prod(int x, int y)` care returnează produsul numerelor întregi  $x$  și  $y$ . Știind că  $a$ ,  $b$  și  $c$  sunt 3 variabile de tip întreg, care dintre expresiile de mai jos atribuie variabilei  $t$  de tip întreg valoarea expresiei  $a*b+a*b*c$ ?

- a) `t = suma(prod(a,b), prod(a,b+c));`
- b) `t = suma(prod(a,b), prod(a,b,c));`
- c) `t = suma(prod(a,b), prod(prod(a,b), c));`
- d) `t = prod(prod(a,b), suma(1, c));`

30. Care dintre următoarele funcții returnează suma cifrelor numărului natural  $n$ ?

```
a) int f(int n)
{
    int s=0;
    while(n!=0)
    {
        s=s+n%10;
        n=n/10;
    }
    return s;
}
```

```
b) int f(int n)
{
    int s=0;
    while(n!=0)
    {
        s=s+n/10;
        n=n%10;
    }
    return s;
}
```

```
c) int f(int n)
{
    int s=0;
    while(n!=0)
    {
        s=s + n%10;
        n=n/10;
    }
}
```

```
    }  
}
```

```
d) int f(int n)  
{  
    int s=0;  
    while(n!=0)  
    {  
        s=n%10;  
        n=n/10;  
    }  
    return s;  
}
```

31. Care dintre următoarele funcții poate fi folosită într-un program pentru a citi de la tastatură un tablou unidimensional format din numere întregi?

- a) `void citire(int v[],int n)`  

```
{  
    scanf("%d",&n);  
    for(int i=0;i<n;i++) scanf("%d",&v[i]);  
}
```
- b) `void citire(int v[],int *n)`  

```
{  
    scanf("%d",n);  
    for(int i=0;i<*n;i++) scanf("%d",&v[i]);  
}
```
- c) `void citire(int *v[],int *n)`  

```
{  
    scanf("%d",&n);  
    for(int i=0;i<n;i++) scanf("%d",&v[i]);  
}
```
- d) `void citire(int *v,int *n)`  

```
{  
    scanf("%d",n);  
    for(int i=0;i<*n;i++) scanf("%d",v+i);  
}
```

32. Care dintre următoarele funcții returnează suma elementelor tabloului unidimensional de numere întregi transmis ca parametru?

- a) `int suma(int v[],int n)`  

```
{  
    int s=0,k=0;  
    while(k<n) s+=v[k++];  
    return s;  
}
```

```
b) int suma(int v[],int n)
{
    int s=0,k=0;
    while(k++<n) s+=v[k];
    return s;
}
```

```
c) int suma(int v[],int n)
{
    int s,k;
    for(k=s=0;k<n;s+=v[k++]);
    return s;
}
```

```
d) int suma(int v[],int n)
{
    int s=0;
    for(int k=n-1;k>=0;k--) s+=v[n-k-1];
    return s;
}
```

33. Funcția `minmax` primește prin parametrul de intrare  $v$  un tablou unidimensional format din numere întregi, iar prin parametrul de intrare  $n$  primește numărul de elemente ale tabloului  $v$ . Funcția trebuie să întoarcă prin doi parametri de ieșire,  $min$  și  $max$ , valoarea minimă și, respectiv, valoarea maximă din tabloul  $v$ . Care dintre următoarele variante reprezintă un antet corect al funcției `minmax`?

- a) `void minmax(int v[],int n,int min,int max)`
- b) `int minmax(int v[],int n,int min,int max)`
- c) `void minmax(int v[],int n,int *min,int *max)`
- d) `void minmax(int *v[],int *n,int *min,int *max)`

34. Fie  $v$  un tablou unidimensional format din 100 de numere reale de tip `double` și numărul natural  $k$  cuprins între 0 și 99. Care dintre următoarele expresii afișează adresa elementului  $v[k]$ ?

- a) `printf("%p",v+k);`
- b) `printf("%p",*(v+k));`
- c) `printf("%p",v+k*sizeof(double));`
- d) `printf("%p",&v[k]);`

35. Fie  $v$  un tablou unidimensional format din 100 de numere reale de tip `double` și  $p$  o variabilă de tip pointer către `double` în care este memorată adresa ultimului element al tabloului  $v$  (`double *p=&v[99];`). Care dintre următoarele expresii afișează numărul de octeți pe care îi ocupă tabloul  $v$  în memorie?

- a) `printf("%d", (p-v+1)*sizeof(double));`

- b) `printf("%d",100*sizeof(double));`
- c) `printf("%d",p-v);`
- d) `printf("%d",sizeof(v));`

36. Care dintre următoarele secvențe de cod poate fi utilizată pentru a alocă dinamic un tablou unidimensional  $a$  format din 100 de numere întregi?

- a) `int *a = (int *)calloc(100*sizeof(int));`
- b) `int *a = (int *)malloc(100*sizeof(int));`
- c) `int *a = (int *)malloc(100);`
- d) `int *a = (int *)calloc(100,sizeof(int));`

37. Care dintre următoarele secvențe de cod poate fi utilizată pentru a alocă dinamic un tablou bidimensional  $a$  format din 10 de linii și 20 de coloane de numere întregi?

- a) `int **a = (int **)malloc(10*sizeof(int *));`  
`for(int i=0;i<10;i++)`  
`a[i]=(int *)malloc(20*sizeof(int));`
- b) `int **a = (int **)calloc(10*sizeof(int *),20*sizeof(int));`
- c) `int *a = (int *)malloc(20*sizeof(int *));`  
`for(int i=0;i<20;i++)`  
`a[i]=(int *)malloc(10*sizeof(int));`
- d) `int **a = (int **)calloc(10,sizeof(int*));`  
`for(int i=0;i<10;i++)`  
`a[i]=(int *)calloc(20,sizeof(int));`

38. Fie  $a$  un tabloul bidimensional pătratic de dimensiune  $n$ . Care dintre următoarele secvențe de cod afișează elementele aflate pe diagonala principală a matricei  $a$ ?

- a) `for(int i=0;i<n;i++)`  
`for(int j=0;j<n;j++)`  
`if(i==j) printf("%d ",a[i][j]);`
- b) `for(int i=0;i<n;i++) printf("%d ",a[i][i]);`
- c) `for(int i=0;i<n;i++)`  
`for(int j=0;j<n;j++)`  
`if(i+j==n-1) printf("%d ",a[i][j]);`
- d) `for(int i=0;i<n;i++) printf("%d ",a[i][n-i-1]);`

39. Fie  $a$  un tabloul bidimensional pătratic de dimensiune  $n$ . Care dintre următoarele secvențe de cod afișează elementele aflate pe diagonala secundară a matricei  $a$ ?

- a) `for(int i=0;i<n;i++)`  
`for(int j=0;j<n;j++)`



## PROGRAMARE PROCEDURALA

```
if(i==j) printf("%d ",a[i][j]);
```

```
b) for(int i=0;i<n;i++) printf("%d ",a[i][i]);
```

```
c) for(int i=0;i<n;i++)  
    for(int j=0;j<n;j++)  
        if(i+j==n-1) printf("%d ",a[i][j]);
```

```
d) for(int i=0;i<n;i++) printf("%d ",a[i][n-i-1]);
```

40. Fie  $a$  un tabloul bidimensional pătratic de dimensiune  $n$ . Care dintre următoarele secvențe de cod afișează elementele triunghiului delimitat de prima coloană, diagonala principală și ultima linie din matricea  $a$ ?

```
a) for(int i=0;i<n;i++)  
    for(int j=0;j<n;j++)  
        if(i>=j) printf("%d ",a[i][j]);
```

```
b) for(int i=0;i<n;i++)  
    for(int j=0;j<n;j++)  
        if(i<=j) printf("%d ",a[i][j]);
```

```
c) for(int i=0;i<n;i++)  
    for(int j=0;j<=i;j++) printf("%d ",a[i][j]);
```

```
d) for(int i=0;i<n;i++)  
    for(int j=i;j<n;j++) printf("%d ",a[i][j]);
```

41. Fie  $a$  un tabloul bidimensional pătratic de dimensiune  $n$ . Care dintre următoarele secvențe de cod afișează elementele triunghiului delimitat de diagonala principală, ultima coloană și prima linie din matricea  $a$ ?

```
a) for(int i=0;i<n;i++)  
    for(int j=0;j<n;j++)  
        if(i>=j) printf("%d ",a[i][j]);
```

```
b) for(int i=0;i<n;i++)  
    for(int j=0;j<n;j++)  
        if(i<=j) printf("%d ",a[i][j]);
```

```
c) for(int i=0;i<n;i++)  
    for(int j=0;j<=i;j++) printf("%d ",a[i][j]);
```

```
d) for(int i=0;i<n;i++)  
    for(int j=i;j<n;j++) printf("%d ",a[i][j]);
```

42. Fie  $a$  un tabloul bidimensional pătratic de dimensiune  $n$ . Care dintre următoarele secvențe de cod afișează suma elementelor de pe fiecare linie a matricei  $a$ ?

- a) 

```
for(int i=0;i<n;i++)
{
    int s=0;
    for(int j=0;j<n;j++)
    {
        s=s+a[i][j];
        printf("%d ",s);
    }
}
```
- b) 

```
int s=0;
for(int i=0;i<n;i++)
{
    for(int j=0;j<n;j++)
    {
        s=s+a[i][j];
        printf("%d ",s);
    }
}
```
- c) 

```
int s=0;
for(int i=0;i<n;i++)
{
    for(int j=0;j<n;j++) s=s+a[i][j];
    printf("%d ",s);
}
```
- d) 

```
for(int i=0;i<n;i++)
{
    int s=0;
    for(int j=0;j<n;j++) s=s+a[i][j];
    printf("%d ",s);
}
```

43. Care dintre următoarele funcții returnează dimensiunea în octeți a unui fișier text a cărui cale este transmisă prin parametrul de intrare *nf*?

- a) 

```
int nb(char *nf)
{
    FILE *f=fopen(nf,"r");
    fseek(f,0,SEEK_END);
    int n=ftell(f);
    fclose(f);
    return n;
}
```
- b) 

```
int nb(char *nf)
{
    char c;
    FILE *f=fopen(nf,"r");
    int n=0;
```

```

while(!feof(f))
{
    fscanf(f,"%c",&c);
    n++;
}
fclose(f);
return n+1;
}

```

```

c) int nb(char *nf)
{
    FILE *f=fopen(nf,"r");
    int n=sizeof(f);
    fclose(f);
    return n;
}

```

```

d) int nb(char *nf)
{
    char s[1001];
    FILE *f=fopen(nf,"r");
    int n=0;
    while(fgets(s,1000,f))
        n++;
    fclose(f);
    return n;
}

```

44. Care dintre următoarele funcții returnează numărul de linii dintr-un fișier text a cărui cale este transmisă prin parametrul de intrare *nf* (se presupune că fișierul nu conține linii vide)?

```

a) int nl(char *nf)
{
    char s[1001];
    FILE *f=fopen(nf,"r");
    int n=0;
    while(fscanf(f,"%s",s)==1)
        n++;
    fclose(f);
    return n;
}

```

```

b) int nl(char *nf)
{
    char c;
    FILE *f=fopen(nf,"r");
    int n=0;
    while(fscanf(f,"%c",&c)==1)
        if(c=='\n') n++;
    fclose(f);
    return n;
}

```

```

    }

```

```

c) int nl(char *nf)
{
    FILE *f=fopen(nf,"r");
    int n=sizeof(f);
    fclose (f);
    return n/sizeof(char *);
}

```

```

d) int nl(char *nf)
{
    char s[1001];
    FILE *f=fopen(nf,"r");
    int n=0;
    while(fgets(s,1000,f))
        n++;
    fclose (f);
    return n;
}

```

45. Considerăm următorul program:

```

#include<stdio.h>
#include<string.h>

int main()
{
    FILE *f=fopen("test.txt","r");
    char s[101],t[101];
    while(fgets(s,100,f))
        strcpy(t,s);
    printf("%s",t);
    fclose(f);
    return 0;
}

```

Știind ca lungimea maximă a unei linii din fișierul text *test.txt* este de 100 de caractere, ce se va afișa după executarea programului de mai sus?

- a) fiecare linie din fișier;
- b) penultima linie din fișier;
- c) ultimul caracter din fișier;
- d) ultima linie din fișier.

46. Considerăm următorul program:

```

#include<stdio.h>
#include<string.h>

```

## PROGRAMARE PROCEDURALA

```
int main()
{
    char s[21],aux[11];
    strcpy(s,"");
    for(int i=1;i<=5;i++)
    {
        printf("Sirul %d: ",i);
        gets(aux);
        .....
    }
    printf("%s",s);
    return 0;
}
```

Presupunând că fiecare dintre cele 5 șiruri care vor fi citite de la tastatură vor fi formate din minim două caractere și maxim 10, stabiliți cu care dintre instrucțiunile de mai jos trebuie înlocuite spațiile punctate din program astfel încât acesta să afișeze șirul format din ultimele două caractere din fiecare dintre cele 5 șiruri citite:

- a) `strcat(s,aux+9);`
- b) `strcat(s,aux[strlen(aux)-1]);`
- c) `strncat(s,aux,strlen(aux)-1);`
- d) `strcat(s,aux+strlen(aux)-2);`

47. Care dintre următoarele secvențe de cod afișează pe ecran șirul de numere 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5?

- a) `for(int i=1;i<=5;i++)`  
`for(int j=1;j<=5;j++)`  
`printf("%d",i);`
- b) `for(int i=1;i<=5;i++)`  
`for(int j=1;j<=i;j++)`  
`printf("%d",i);`
- c) `for(int i=1;i<=5;i++)`  
`for(int j=1;j<=i;j++)`  
`printf("%d",j);`
- d) `for(int i=1;i<=4;i++)`  
`for(int j=i+1;j<=5;j++)`  
`printf("%d",i);`

48. Stabiliți care dintre următoarele funcții întorc poziția primei valori strict pozitive din tabloul  $v$  format din  $n$  numere întregi sau -1 dacă tabloul nu conține nici un număr pozitiv:

- a) `int p(int v[],int n)`  
`{`  
`int i,x=-1;`

## PROGRAMARE PROCEDURALA

```
    for(i=0;i<n;i++)
        if(v[i]>0) x=i;
    return x;
}
```

b) `int p(int v[],int n)`  
{  
 int x=0;  
 while(v[x]<=0) x++;  
 return x-1;  
}

c) `int p(int v[],int n)`  
{  
 int i,x=-1;  
 for(i=0;i<n;i++)  
 if((v[i]>0)&&(x<0)) x=i;  
 return x;  
}

d) `int p(int v[],int n)`  
{  
 int i;  
 for(int i=0;i<n;i++)  
 if (v[i]>0) return i;  
 return -1;  
}

49. Considerăm următoarea secvență de cod:

```
int i=0,j;  
while (i+j<=10)  
{  
    i++;  
    j=j-2;  
}
```

Valoarea minimă posibilă pentru variabila  $j$  astfel încât instrucțiunea repetitivă de mai sus să nu fie o ciclare infinită este:

- a) 8
- b) 10
- c) 12
- d) 17

50. Considerăm următorul program:

## PROGRAMARE PROCEDURALA

```
#include<stdio.h>

void p(int v[],int *n)
{
    int i,j,g;
    do
    {
        g=0;
        for(i=0;i<*n;i++)
            if(v[i]<0)
            {
                for(j=i;j<*n-1;j++) v[j]=v[j+1];
                (*n)--;
                g=1;
            }
    }
    while(g);
}

int main()
{
    int i,v[]={-1,2,-3,-4,5},n=5;
    p(v,&n);
    for(i=0;i<n;i++)
        printf("%d ",v[i]);
    return 0;
}
```

Ce valori vor fi afișate pe ecran după executarea programului de mai sus?

- a) -1 -3 -4
- b) 2 5**
- c) -1 2 -3 -4
- d) 2 -3 -4 5

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Programare orientata pe obiecte (C++)**



1. Fie secvența:

```
class cls{
public:
    cls(){ cout<<"constructor";}
    cls(cls &c){cout<<"constructor de copiere";}
};
int f(cls c){ return 1;}
int main(){
    cls c;
    f(c);
    return 0;
}
```

În momentul executării programului de mai sus:

- a) constructorul de clasă se apelează o dată, iar cel de copiere nu se apelează;
- b) constructorul de clasă și cel de copiere se apelează fiecare câte o dată;**
- c) constructorul de copiere se apelează o dată, iar cel de clasă nu se apelează;
- d) constructorul de clasă se apelează de două ori, iar cel de copiere nicio dată;
- e) constructorul de clasă și cel de copiere se apelează fiecare de câte două ori.

2. Fie secvența:

```
class cls{
public:
    cls(){ cout<<"constructor";}
    cls(cls &c){cout<<"constructor de copiere";}
};
int f(cls &c){ return 1;}
int main(){
    cls c;
    f(c);
    return 0;
}
```

În momentul executării programului de mai sus:

- a) constructorul clasei se apelează o dată, iar cel de copiere nu se apelează nicio dată;**
- b) constructorul de clasă și cel de copiere se apelează fiecare câte o dată;
- c) constructorul de copiere se apelează o dată, iar cel de clasă nu se apelează;
- d) constructorul clasei se apelează de două ori, iar cel de copiere nicio dată;
- e) constructorul de clasă și cel de copiere se apelează fiecare de câte două ori.

3. Fie secvența:

```
class C{
    int a;
public:
    virtual void metoda1()=0;
    virtual void metoda2()=0;
};
```

```
int main(){
    C *pob;           //declarația 1
    C ob;             //declarația 2
    C *vpob[5];       //declarația 3
    C vob[5];         //declarația 4
    return 0;
}
```

Declarațiile admise în acest caz sunt:

- a) Declarațiile 1 și 2;
- b) **Declarația 1;**
- c) Declarațiile 2 și 4;
- d) **Declarația 3;**
- e) Declarațiile 1, 2 și 3.

4. Fie clasa :

```
class c{
    int a, b ;
public :
    c (int , int ) ;
    int det_a ( ) {return a ;}
    ~c ( ) ;
};
```

Semnul ~ are rolul :

- a) de a nega pe biți rezultatul returnat de metoda c ( );
- b) **de a preciza existența destructorului;**
- c) de a nega logic rezultatul returnat de metoda c ( );
- d) de a supraîncarca constructorul clasei;
- e) de a supraîncarca operatorul ~

5. Secvența următoare:

```
class c1{
public:
    int a;
    c1(int y){ a=y;cout<<"constructor 1";}
    ~c1(){cout<<"destructor 2";}
};
class c2:public c1{
public:
    int b;
    c2(int y, int x):c1(y) { b=x; cout<<"constructor 2";}
    ~c2(){cout<<"destructor 2";}
};
int main(){
    c1 ob1(2);
    c2 ob2(2,3);
    return 0;
}
```

```
}
```

afișează:

- a) constructor 1 constructor 2 destructor 2 destructor 1
- b) constructor 1 constructor 1 constructor 2 destructor 2 destructor 1 destructor 1**
- c) constructor 1 constructor 2 constructor 1 destructor 1 destructor 2 destructor 1
- d) constructor 1 constructor 1 constructor 2 destructor 2 destructor 1

6. Fie următorul program C++:

```
#include <iostream.h>
class B{
    public:
        B() {cout<<"B() "<<endl;}
        ~B() {cout<<"~B() "<<endl;}
};

class D: public B{
    public:
        D() {cout<<"D() "<<endl;}
        ~D() {cout<<"~D() "<<endl;}
};

int main(){
    B *b=new B();
    delete b;
    b=new D();
    delete b;
    return 0;
}
```

Programul afișează:

- a) B() ~B() B() D() ~D()
- b) B() ~B() B() D() ~B()**
- c) B() ~B() B() ~B()
- d) B() ~B() D() ~B()

7. Fie programul:

```
#include <iostream.h>
class B{
    public:
        B() {cout<<"B() "<<endl;}
        B(B &b) {cout<<"B(B &b) "<<endl;}
};

class D: public B{
    public:
        D() {cout<<"D() "<<endl;}
        D(D &d) {cout<<"D(D &d) "<<endl;}
};

int main(){
    B b;
```

```

    B b1(b);
    D d;
    D d1(d);
    return 0;
}

```

Programul afișează:

- a) B() B(B&b) B() D() B(B &b) D(D &d)
- b) B() B() B(B&b) B() D() B(B &b) D() B(B &b)
- c) B() B(B&b) D() B(B &b) D() B(B &b)
- d) **B() B(B&b) B() D() B() D(D &d)**

8. Fie clasa :

```

class c {
    int a,b;
public:
    float c (int, int)
    int get_a {return a;}
    c ();
};

```

Declarația `float c(int, int)` ar putea corespunde unui constructor al clasei?

- a) da, fiind o supraîncărcare a celui existent;
- b) nu, deoarece crează ambiguitate;
- c) **nu, deoarece constructorul nu are tip returnat;**
- d) nu, deoarece nu este de tip friend.

9. Fie secvența următoare:

```

class persoana{
    int varsta;
public:
    persoana(int v=18){varsta=v;}
    persoana& operator++(int){varsta++; return *this;}
    int get_varsta(){return varsta;}
};
int main(){
    persoana p(20);
    cout<<p++.get_varsta();
    return 0;
}

```

Secvența afișează:

- e) **21**
- f) 20
- g) 18
- h) 19

10. O funcție declarată friend în clasa de bază:

- a) rămâne friend în clasa derivată, pentru partea moștenită;
- b) are acces pe toată clasa derivată;
- c) nu are acces pe zonele private și protected ale clasei derivate;
- d) nu are acces pe zona private a clasei derivate;
- e) are acces pe zonele public și protected ale clasei derivate.

11. Se consideră următoarea secvență de program:

```
class B{
    private:
        int x,y;
    public:
        B(int a,int b){ x=a;y=b; }
        B(const B &a){ x=a.x; y=a.y;}
};
```

În care dintre următoarele situații se realizează copierea unui obiect într-altul:

- a) B c1(4,5);
- b) B c2(0.0, 0,0);
- c) B c1, c3=c1;
- d) B c4(1);
- e) B c1, c5(c1).

12. Fie următorul program:

```
#include<iostream.h>
class cls {
    static int i;
    int j;
    public:
        cls(int x=7) { j=x; }
        static int imp(int k){ cls a; return i+k+a.j; } };
int cls::i;
int main()
{ int k=5;
  cout<<cls::imp(k);
  return 0;
}
```

Indicați ce se va afișa pe ecran în urma executării programului:

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

13. Fie următorul program:

```
#include <iostream.h>
class B{
```

## PROGRAMARE IN C++

```
public:
    virtual void f() { cout<<"B::f() ";}
    void g() { cout<<"B::g() ";}
};

class D: public B{
public:
    void f() { cout<<"D::f() ";}
    void g() { cout<<"D::g() ";}
};

int main(){
    int i;
    B *a=new B();
    B *b=new D();
    a->f(); b->f();
    a->g(); b->g();
    return 0;
}
```

Indicați ce se va afișa pe ecran în urma executării programului:

- a) D::f() B::f() B::g() B::g() B::g()
- b) B::f() D::f() B::g() B::g()**
- c) B::f() D::f() B::g() D::g()
- d) B::f() B::g() D::f() D::g()

14. Fie următorul program:

```
#include <iostream.h>
class B{
public:
    virtual void f() { cout<<"B::f() ";}
    void g() { cout<<"B::g() ";}
};

class D1: public B{
public:
    void f() { cout<<"D1::f() ";}
    void g() { cout<<"D1::g() ";}
};

class D2: public B{
public:
    void g() { cout<<"D2::g() ";}
};

int main(){
    int i;
    B *a=new B();
    B *b=new D1();
    B *c=new D2();
    a->f(); b->f(); c->f();
    a->g(); b->g(); c->g();
    return 0;
}
```

Indicați ce se va afișa pe ecran în urma executării programului:

- e) B::f() D1::f() B::f() B::g() B::g() B::g()
- f) D2::f() D1::f() B::f() B::g() B::g() B::g()
- g) B::f() D1::f() D::f() B::g() D1::g() D2::g()
- h) B::f() D1::f() B::f() B::g() D1::g() D2::g()
- i) B::f() B::f() D2::f() B::g() B::g() D2::g()

15. Fie următorul program:

```
#include<iostream.h>
class salariat{
    int varsta;
    public:
        salariat (int v=20) {varsta =v;}
        operator int() { return varsta;}
        salariat& operator++(){varsta++; return *this;}
        salariat operator++ (int) { varsta++; return *this;}
};
int main(){
    salariat s(21);
    int a=s++, b=++s;
    cout<<a<<"    "<<b<<endl;
    return 0;
}
```

Programul afișează:

- a) 20 21
- b) 21 22
- c) 22 23
- d) 20 22
- e) 21 23

16. Fie următorul program:

```
#include <iostream.h>
class Cerc{
    public:
        float raza;
        Cerc(float r){raza=r;}
        float get_raza(){return raza;}
        Cerc operator++(){raza++; return *this;}
        Cerc operator--(){raza--; return *this;}
};
int main(){
    Cerc c(3.5);
    cout<<((++(++c)).get_raza())<<" ";
    cout<<c.get_raza()<<" ";
    cout<<((--(--c)).get_raza())<<" ";
    cout<<c.get_raza()<<" ";
    return 0;
}
```

}

Programul afișează:

- a) 3.5 4.5 2.5 3.5
- b) 5.5 4.5 2.5 2.5
- c) 2.5 5.5 4.5 3.5
- d) 5.5 4.5 2.5 3.5
- e) 4.5 2.5 3.5 5.5

17. O metodă statică a unui obiect se caracterizează prin faptul că:

- a) nu primește pointerul la obiect `this`;
- b) folosește numai datele publice;
- c) se poate apela prin numele clasei;
- d) nu poate fi definită decât inline;
- e) dacă prelucrează obiecte, primește obiectele ca parametrii expliți.

18. Fie secvența de program:

```
#include <iostream.h>
class C{
    public:
        static int s;
};
int C::s=0;
int main(){
    int a=7; C::s=a;
    cout<<C::s;
    return 0;
}
```

În secvența de mai sus, inițializarea lui `s` este:

- a) ilegală, deoarece nu există niciun obiect creat;
- b) ilegală, deoarece `s` este inițializat în afara clasei;
- c) ilegală, deoarece `s` este dublu definit, în clasă și în afara ei;
- d) ilegală, deoarece datele statice pot fi doar private;
- e) corectă, deoarece membri statici există înainte de a se crea obiecte din clasă.

19. Fie secvența:

```
class complex{
    double re;
    double im;
    public:
        complex(double x=1.0, double y=6.80){re=x; im=y;}
        complex( const complex &u){re=u.re; im=u.im;}
};
```

Precizați în ce situație se utilizează constructorul de copiere:



- a) `complex z1(5.2, 3.6);`
- b) `complex z1(5.2, 3.6), z2=z1;`
- c) `complex z3(0.1,1.0);`
- d) `complex z1(5.2, 3.6), z4(z1);`
- e) `complex z5(-0.1,28.7).`

20. Fie secvența :

```
class A1{
    public:
        A1(){cout << "A1 "};
};
class A2{
    public:
        A2(){cout << "A2 "};
};
class AA1 : public A1, virtual public A2{
    public:
        AA1(){cout << "AA1 "};
};
class AA2 : public A1, virtual A2{
    public:
        AA2(){cout << "AA2 "};
};
class B : public AA1, virtual public AA2{
    public:
        B(){cout << "B "};
};
int main(){
    B obl;
    return 0;
}
```

Secvența afișează:

- a) A1 A2 AA2 A1 AA1 B
- b) A2 A2 AA2 AA1 A1 B
- c) A1 A2 AA2 A1 B AA1
- d) A2 A1 AA2 A1 AA1 B
- e) A2 A1 A2 AA1 A1 B

21. Care dintre afirmațiile următoare sunt adevărate?

- a) precedența unui operator poate fi modificată prin redefinire;
- b) aritatea unui operator nu poate fi modificată prin redefinire;
- c) asociativitatea unui operator poate fi modificată prin redefinire;
- d) semnificația modului în care lucrează un operator asupra obiectelor de tipuri predefinite nu poate fi schimbată prin redefinire.

22. Care dintre afirmațiile următoare sunt adevărate?

- a) funcțiile inline nu pot fi funcții virtuale;
- b) constructorii pot fi funcții virtuale;
- c) orice funcție membru statică este funcție virtuală;
- d) destructorul poate fi funcție virtuală.

23. Fie programul:

```
#include <iostream.h>
class Cerc{
    float raza;
public:
    Cerc(float r){raza=r;}
    float get_raza(){return raza;}
    void operator++(){raza++;}
};
class Cilindru : public Cerc{
    float inaltime;
public:
    Cilindru(float raza, float i):Cerc(raza){inaltime=i;}
    void operator++(){inaltime++;}
    float get_inaltime(){return inaltime;}
};
int main(){
    Cerc *pc;
    Cilindru c(2,6);
    pc=&c;
    ++ *pc;
    cout<<pc->get_raza()<<" "<<c.get_inaltime()<<endl;
    return 0;
}
```

Programul afișează:

- a) 2 5
- b) 2 6
- c) 3 6
- d) 2 5

24. Care dintre afirmațiile următoare sunt false?

- a) obiectele unei clase derivate au acces la membrii privați ai clasei sale de bază;
- b) relația de moștenire este tranzitivă;
- c) funcțiile friend ale clasei de bază se moștenesc de către clasa derivată;
- d) constructorul și destructorul clasei de bază se moștenesc în clasa derivată.

25. Fie următorul program:

```
#include<ostream.h>
class persoana{
    int varsta, salariul;
    friend ostream & operator<<(ostream &out,persoana p){
        out<<p.varsta<<" "<<p.salariul; return out;
    }
}
```

```

        public:
            persoana(int v){varsta=v;salariul=0;}
            persoana() {varsta=0;salariul=0;}
};
int main(){
    persoana p(1);cout<<p;
    return 0;
}

```

Programul afișează:

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

26. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau prin funcții friend. Diferența dintre aceste două posibilități constă în:

- a) lista de parametri;
- b) obiectul returnat;
- c) precedența operatorilor;
- d) aritatea operatorului.

27. Fie programul:

```

class c{
    int a;
    public:
        c(){};
        c(const c&){};
        void operator=(c&){};
};
int main(){
    c a;
    c b=a;
}

```

Linia de cod `c b=a;` determină:

- a) executarea constructorului de copiere;
- b) executarea metodei prin care se supraîncărcă operatorul `=`;
- c) executarea atât a constructorului de copiere, cât și a metodei operator `=`;
- d) o eroare, deoarece nu este permisă combinarea atribuirii cu o declarație;
- e) executarea constructorului implicit.

28. Fie următorul program:

```

#include<iostream.h>
class cls{
    public:
        ~cls(){cout<<"\n Destructor";}
}

```

```
};
int main() {
    cls *po=new cls[3];
    delete []po;
}
```

Destructorul clasei:

- a) nu se apelează nicio dată;
- b) se apelează o dată;
- c) se apelează de trei ori;
- d) se apelează de patru ori.

29. O funcție independentă declarată friend în domeniul public dintr-o clasă și care primește ca parametru o referință la un obiect al clasei respective are acces:

- a) doar la membrii declarați public;
- b) la toți membrii;
- c) la membrii public și la cei protected;
- d) la membrii protected.

30. Fie următorul program:

```
#include<iostream.h>
class A{
    int a[3];
    public:
        A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
        int& operator[] (int i){return a[i];}
};
int main() {
    A ob(1,2,3); cout<<ob[1];
    ob[1]=25; cout<<ob[1];
    return 0;
}
```

Ce se poate afirma despre operator[]()?

- a) produce supraîncărcarea unei funcții;
- b) produce supraîncărcarea unui operator unar;
- c) supraîncărcă operatorul [];
- d) este o funcție membru oarecare a clasei A, care nu produce supraîncărcarea unui operator;
- e) reprezintă un operator ternar.

31. Considerăm următorul program:

```
#include<iostream.h>
class C{
    public:
        int x;
        C(int v) { x=v;}
        double operator+(C &c, double d){return c.x+d;}
```

## PROGRAMARE IN C++

```
double operator+(double d, C &c){return c.x+d;}
};
int main() {
    C c(5);
    cout<<2+c+3;
    return 0;
}
```

Stabiliți care dintre următoarele afirmații sunt adevărate:

- a) supraîncărcările operator + () trebuie să fie friend;
- b) supraîncărcările operator+() nu se justifică deoarece au același cod;
- c) programul afișează 10;
- d) supraîncărcările operator+() trebuie să returneze referințe.

32. Fie programul:

```
#include<iostream.h>
class c1{ int a;};
class c2:public c1{
public:
    int b;
    void scrie_a( ) { cout<<a; }
};
int main(){
    c2 ob; ob.scrie_a();
    return 0;
}
```

Selectați afirmația corectă:

- a) funcția scrie\_a() nu are acces asupra unui membru privat;
- b) programul afișează valoarea lui a;
- c) derivarea publică este incorect realizată;
- d) prin derivare publică, accesul la membrii moșteniți devine public.

33. Fie programul următor:

```
#include<iostream.h>
class B{
    int x;
public:
    B(int i=10) { x=i;}
    int get_x() { return x; }
};
class D: public B{
public:
    D(int i):B(i){}
    D operator+(const D& a) {return x+a.x; }
};
int main(){
    D ob1(7), ob2(-12);
    cout<<(ob1+ob2).get_x();
    return 0;
}
```

```
}
```

Programul afișează:

- a) eroare, clasa B nu poate fi moștenită de clasa D;
- b) eroare, metoda operator nu are acces la un membru privat al clasei de bază;
- c) programul afișează valoarea -5;
- d) eroare, operatorul + nu se poate aplica pentru tipuri abstracte de date.

34. Fie următorul program:

```
#include<iostream.h>
class B1{int x;};
class B2{int y;};
class B3{int z;};
class B4{int t;};
class D: public B1, private B2, protected B3,B4 {public : int m;};
int main(){
    D d;
    cout<<d.m;      //varianta 1
    cout<<d.x;      //varianta 2
    cout<<d.y;      //varianta 3
    return 0;
}
```

Variantele care permit accesul la variabile pentru afișare sunt:

- a) 1+3;
- b) 1+2;
- c) 1+2+3;
- d) 1

35. Considerăm următorul program:

```
class vector{
    int * pe, nr_c;
public:
    operator int () {return nr_c;}
    vector(int);
};
vector::vector(int n){
    pe=new int[n]; nr_c=n;
    while(n-->0) pe[n]=n;
}
void f(int i){cout<<i<<endl;}
int main(){
    vector x(10);
    f(x);
    return 0;
}
```

Programul afișează:

- a) 9
- b) 10
- c) numerele de la 1 la 10
- d) numerele de la 0 la 9

36. Considerăm următorul program:

```
class c{
    int a;
    public:
        virtual void metoda1()=0;
        virtual void metoda2(int)=0;
};
int main{
    c *pob;           //declarația 1
    c ob;             //declarația 2
    c *vpob[3];       //declarația 3
    c vob[3];         //declarația 4
    return 0;
}
```

Declarațiile admise:

- a) 1+2;
- b) 1+2+3+4
- c) nici una
- d) 1+3;

37. Fie data următoarea ierarhie:

```
class B {... }
class D1:B{...}
class D2:B{...}
class M1:D1, public D2{...}
class M2:virtual D1, virtual D2 {...}
```

Considerăm următoarele afirmații:

1. clasa M1 va moșteni un obiect de tip B;
2. clasa M1 va moșteni două obiecte de tip B;
3. clasa M2 va va moșteni un obiect de tip B;
4. clasa M2 va moșteni două obiecte de tip B.

Precizați care dintre afirmațiile de mai sus sunt corecte:

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

38. Fie următorul program:

## PROGRAMARE IN C++

```
#include<iostream.h>
class B{
    public:
        int x;
        B(int i=16) { x=i; }
        B f(B ob) { return x+ob.x; }
};
class D: public B{
    public:
        D(int i=25) { x=i; }
        B f(B ob) { return x+ob.x+1; }
        void afisare(){ cout<<x; }
};
int main(){
    B *p1=new D, *p2=new B, *p3=new B(p1->f(*p2));
    cout<<p3->x;
    return 0;
}
```

Programul afișează:

- a) 41
- b) eroare, nu se poate instanția un obiect al unei clase derivate printr-un pointer la un obiect de tip clasa de bază;
- c) 44
- d) 45

39. Fie următorul program:

```
#include<iostream.h>
class B{
    int i;
    public:
        static int x;
        B() { x++; i=1; }
        ~B() { x--; }
        static int get_x() { return x; }
        int get_i() { return i; }
};
int B::x;
class D: public B{
    public:
        D() { x++; }
        ~D() { x--; }
};
int f(B *q){ return (q->get_i())+1;}
int main(){
    B *p=new B;
    cout<<f(p);
    delete p;
    p=new D;
    cout<<f(p);
    delete p;
    cout<<D::get_x();
    return 0;
}
```



Programul afișează:

- a) eroare, data membră statică x nu este inițializată;
- b) eroare, metoda get\_x() nu poate fi declarată static;
- c) **programul afișează valoarea 221;**
- d) programul afișează valoarea 220.

40. Fie următorul program:

```
#include <iostream.h>
template<class T, class E>
float f(T x, E y){ return x+y;}
float g(int x, float y){ return x-y;}
int main(){
    int a=5;
    float b=8.6;
    cout<<g(a,b);
    return 0;
}
```

Programul afișează:

- a) 3
- b) eroare, parametrizarea clasei T este incorrect realizată
- c) 13.6
- d) **-3.6**

41. Fie următorul program:

```
#include <iostream.h>
template<class T>
int f(T x, T y){ return x+y;}
int f(int x, int y){ return x-y;}
int main(){
    int a=5;
    float b=8.6;
    cout<<f(a,b);
    return 0;
}
```

Programul afișează:

- e) **-3**
- f) eroare, parametrizarea clasei T este incorrect realizată
- g) 13.6
- h) 3.6

42. Fie următorul program:

```
#include<iostream.h>
class B{
    int x;
    public:
```

## PROGRAMARE IN C++

```
B(int i=10) { x=i; }
int get_x() { return x; };
class D: public B{
public:
    D(int i):B(i) {}
    D operator+(const D& a) {return x+a.x; };
int main()
{ D ob1(7), ob2(-12);
  cout<<(ob1+ob2).get_x();
  return 0;
}
```

Indicați ce se va afișa pe ecran în urma executării programului:

- a) -5
- b) -4
- c) eroare, în clasa derivată D nu se poate accesa data membră privată x a clasei B
- d) -3

43. Fie următorul program:

```
#include<iostream.h>
class B{
public:
    int x;
    B(int i=16) { x=i; }
    B f(B ob) { return x+ob.x; } };
class D: public B{
public:
    D(int i=25) { x=i; }
    B f(B ob) { return x+ob.x+1; }
    void afisare(){ cout<<x; } };
int main()
{
    B *p1=new D, *p2=new B, *p3=new B(p1->f(*p2));
    cout<<p3->x;
    return 0;
}
```

Programul afișează:

- a) 41
- b) eroare, în clasa derivată D nu se poate accesa data membră x a clasei B
- c) 16
- d) 25

44. Fie următorul program:

```
#include<iostream.h>
class cls1{
public:
    int a;
    cls1() { a=7; }
};
class cls2{
public:
```

```

        int b;
        cls2(int i) { b=i; }
        cls2(cls1& x) { b=x.a; }
};
int main(){
    cls1 x;
    cout<<x.a;
    cls2 y(x);
    cout<<y.b;
    return 0;
}

```

Programul afișează:

e) 77

- f) eroare, constructorul de copiere nu este corect definit
- g) eroare, constructorul de copiere nu poate accesa o dată publică a clasei cls1
- h) 78

45. O funcție friend diferă de o metodă obișnuită a unei clase prin faptul că:

- a) nu se poate defini inline;
- b) **nu primește pointerul implicit la obiect this;**
- c) nu poate accesa decât partea publică a obiectului;
- d) se folosește doar pentru supraîncărcarea operatorilor;
- e) nu poate returna valori.

46. O funcție independentă declarată friend în domeniul private dintr-o clasă și care primește ca parametru o referință la un obiect al clasei respective are acces:

- a) doar la membrii publici;
- b) **la toti membrii;**
- c) la membrii public și protected;
- d) la membrii private;
- e) la toti membrii, dar îl poate doar consulta, nu și modifica.

47. O funcție independentă declarată friend în domeniul public dintr-o clasă și care primește ca parametru o referință la un obiect al clasei respective are acces:

- a) doar la membrii publici;
- b) **la toti membrii;**
- c) la membrii public și protected;
- d) la membrii private;
- e) la toti membrii, dar îl poate doar consulta, nu și modifica.

48. Fie următorul program:

```

#include <iostream>
using namespace std;
class c{
    int a;

```

## PROGRAMARE IN C++

```
public :
    c() {}
    c(const c&);
    c& operator =(c&);};
c& c::operator=(c &c){ cout << endl << "copiere cu egal"; return c;}
c::c(const c &c) { cout << endl << "Constructor de copiere"; }
int main()
{
    c x,y=x;
    c b=x; x=y;
};
```

Programul:

- a) apeleaza de doua ori operator=(), o data constructorul de copiere si o data constructorul implicit;
- b) apeleaza de trei ori constructorul de copiere, o data constructorul implicit;
- c) apeleaza de trei ori supraincercarea operatorului =;
- d) apeleaza de doua ori constructorul de copiere si de trei ori operator=();
- e) **apeleaza de doua ori constructorul de copiere, o data operator=() si o data constructorul implicit;**

49. De câte ori este apelat destructorul clasei Persoana în programul următor?

```
#include <iostream>
using namespace std;
class Persoana{
public:
    Persoana() {cout<<"Constructor"<<endl;}
    ~Persoana() {cout<<"Destructor"<<endl;};};
int main(){
    Persoana** ppp;
    ppp = new Persoana*[5];
    for(int i=0; i<5; i++)
        ppp[i] = new Persoana();
    for(int i=0; i<5; i++)
        delete ppp[i];
}
```

Răspuns:

- a. 10;
- b. 6;
- c. 7;
- d. 5;**
- e. niciunul din răspunsurile anterioare.

50. În programul următor:

```
#include <iostream>
using namespace std;
class Persoana{
    int varsta;
```

## PROGRAMARE IN C++

```
char* nume;
public:
    Persoana(int v=0, char* n="Oarecare"):varsta(v){
        this->nume = new char[strlen(n)+1];
        strcpy(this->nume,n);
        cout<<"Constructor"<<endl;}
    Persoana(Persoana& p){
        this->varsta = p.varsta;
        this->nume = new char[strlen(p.nume)+1];
        strcpy(this->nume, p.nume);
        cout<<"Constructor de copiere"<<endl;}
    void operator=(Persoana& p){
        this->varsta = p.varsta;
        delete[] this->nume;
        this->nume = new char[strlen(p.nume)+1];
        strcpy(this->nume, p.nume);
        cout<<"Operator="<<endl;}
    ~Persoana(){ cout<<"Destructor"<<endl;}};

int main()
{
    Persoana p1, p2(20, "Gigel");
    Persoana p3 = p1;
    p3 = p2;
    Persoana p4 = p1;
}
```

Sunt apelate următoarele:

- a. constructor – de patru ori, constructor de copiere – o dată, destructor – de patru ori;
- b. constructor – de trei ori, constructor de copiere - de două ori, destructor de cinci ori;
- c. constructor – de două ori, constructor de copiere – de două ori, operator= - o dată, destructor – de patru ori;
- d. constructor – de două ori, constructor de copiere – o dată, operator= - de două ori, destructor – de două ori;
- e. constructor – de două ori, constructor de copiere – o dată, operator= - de două ori, destructor – de patru ori.

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Programare in Java**

**1. Fie următoarea clasă Java:**

```
class C
{
    int a;
    float x;
    boolean b;
}
```

Stabiliți care dintre următoarele instrucțiuni este corectă:

- a) `C ob = new C(1);`
- b) `C ob = new C(1,1.0);`
- c) `C ob = new C();`
- d) `C ob = new C(1,1.0,true);`

**2. Fie următorul program Java:**

```
class C
{
    public static int a=1;
}

public class test
{
    public static void main(String[] args)
    {
        C ob=new C();
        C.a++;
        ob.a++;
        System.out.println(C.a);
    }
}
```

După executarea programului, va fi afișată valoarea:

- a) 3;
- b) 2;
- c) 1;
- d) nicio valoare, deoarece programul este incorect sintactic și nu va putea fi executat.

**3. Fie următorul program Java:**

```
class C{public static int a=1;}

public class teste_grila
{
    public static void main(String[] args)
    {
        C ob1=new C();
        C ob2=new C();
        ob1.a++;
        System.out.println(ob2.a);
    }
}
```

```
}

```

După executarea programului, va fi afișată valoarea:

- a) 0;
- b) 2;**
- c) 1;
- d) nicio valoare, deoarece programul este incorect sintactic și nu va putea fi executat.

4. Un program Test scris în limbajul Java poate fi compilat folosind comanda:

- a) javac Test
- b) java Test.java
- c) javac Test.class
- d) javac Test.java**

5. Un program Test scris în limbajul Java și compilat, poate fi rulat folosind comanda:

- a) javac Test.java
- b) java Test**
- c) java Test.class
- d) java Test.java

6. În Java o clasă poate extinde:

- a) cel mult o interfață
- b) oricâte clase
- c) cel mult o clasă**
- d) oricâte interfețe

7. În Java o interfață poate extinde:

- a) cel mult o interfață
- b) oricâte interfețe**
- c) cel mult o clasă
- d) oricâte clase

8. În Java o clasă poate implementa:

- a) o clasă
- b) oricâte clase
- c) o interfață
- d) oricâte interfețe**

9. Fie următorul program Java:

```
class A
{
    public A() { System.out.print("A"); }
}

class B extends A
{

```



```

    public B() { System.out.print("B"); }
}

class C extends B
{
    public C() { System.out.println("C"); }
}

public class test
{
    public static void main(String[] args)
    {
        C ob=new C();
    }
}

```

După executarea programului, se va afișa:

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

**10.** Fie următorul program Java:

```

class A
{
    public int x=1;
    public A() { x++; }
}

class B extends A
{
    public B() { x++; }
}

class C extends B
{
    public int x=1;
    public C() { x++; }
}

public class test
{
    public static void main(String[] args)
    {
        B b=new B();
        C c=new C();
        System.out.println(b.x+" "+c.x);
    }
}

```

După executarea programului, se va afișa:

- a) 3 4
- b) 3 2

- c) 2 2
- d) 3 3

**11.** Fie următorul program Java:

```
class A
{
    int x=0;
    public A(int n) { x=n; }
}

class B extends A
{
    int x=1;
    public B(int n) { super(n); }
}

public class test
{
    public static void main(String[] args)
    {
        A a=new A(5);
        B b=new B(7);
        System.out.println(a.x+" "+b.x);
    }
}
```

După executarea programului, se va afișa:

- a) 05
- b) 51**
- c) 57
- d) 01

**12.** Fie următorul program Java:

```
class A{
    static void staticMethod() {
        System.out.println("Metoda statica");
    }
}

public class Test{
    public static void main(String[] args){
        A a = null;
        a.staticMethod();
    }
}
```

După executarea programului, se va afișa:

- a) programul afișează mesajul Metoda statica
- b) se obține o eroare la compliere cauzată de faptul ca o metodă statică nu poate fi invocată folosind o referință
- c) se obține la executare excepția `java.lang.NullPointerException`

**13. Program următor:**

```

class A
{
    int x=10;
    static int y=20;
}
class B extends A
{
    int x=30;
    static int y=40;
}
public class Test
{
    public static void main(String[] args) {
        A ob = new B();
        System.out.println(ob.x+" "+ob.y);
    }
}

```

Afișează

- a) 10 20
- b) 30 40
- c) 30 20
- d) eroare la executare

**14. Fie următoarele declarații în Java:**

```

interface Patrat
{
    public float aria();
    public float perimetru();
}

class Patrat_1 implements Patrat
{
    float L;

    public Patrat_1(float x) { L=x; }
    public float aria() { return L*L; }
}

class Patrat_2 extends Patrat_1 implements Patrat
{
    public Patrat_2(float L) { this.L=L; }
    public float perimetru() { return 4*L; }
}

```

Stabiliți care dintre următoarele propoziții sunt adevărate:

- a) definiția clasei Patrat\_1 este incorectă deoarece nu implementează metoda perimetru a interfeței Patrat;
- b) constructorul clasei Pătrat\_2 este incorect deoarece nu are acces la pointerul this;
- c) constructorul clasei Pătrat\_2 este incorect deoarece nu are apelează constructorul superclasei Patrat\_1;

- d) definiția clasei `Patrat_2` este incorectă deoarece nu implementează metoda `aria` a interfeței `Patrat`.

### 15. Programul următor:

```
public class Test {
    static void test(int a[]){
        a[0] = 100;
        a = new int[]{10,20,30,40,50};
        a[1] = 200;
        System.out.println(Arrays.toString(a));
    }

    public static void main(String[] args){
        int []v = {1,2,3,4,5,6,7};
        test(v);
        System.out.println(Arrays.toString(v));
    }
}
```

Afișează

- a) [10, 200, 30, 40, 50]  
[1 2, 3, 4, 5, 6, 7]
- b) [10, 200, 30, 40, 50]  
[100, 2, 3, 4, 5, 6, 7]
- c) [10, 20, 30, 40, 50]  
[100, 2, 3, 4, 5, 6, 7]
- d) [10, 20, 30, 40, 50]  
[100, 2, 3, 4, 5, 6, 7]

### 16. Fie următorul program Java:

```
interface Patrat
{
    public float A();
    public float P();
}

interface Dreptunghi
{
    public float A();
    public float P();
}

class Patrulater_1 implements Patrat,Dreptunghi
{
    float L;

    public Patrulater_1(float x) { L=x; }
    public float A() { return L*L; }
    public float P() { return 4*L; }
}

class Patrulater_2 implements Patrat, Dreptunghi
```

```

{
    float L,l;

    public Patrulater_2(float x, float y) { L=x; l=y; }
    public float A() { return L*l; }
    public float P() { return 2*(L+l); }
}

public class teste_grila
{
    public static void main(String[] args)
    {
        Dreptunghi d = new Patrulater_1(10);
        Patrat p = new Patrulater_2(10,20);

        System.out.println(d.A()+" "+d.P()+" "+p.A()+" "+p.P());
    }
}

```

Stabiliți care dintre următoarele propoziții sunt adevărate:

- a) programul este incorect deoarece apare un conflict de nume pentru ca în interfețele Patrat și Dreptunghi sunt definite metode cu aceeași semnătură, iar clasele Patrulater\_1 și Patrulater\_2 implementează fiecare ambele interfețe;
- b) programul este incorect deoarece în interfețele Patrat și Dreptunghi sunt definite metodele A și P cu aceeași semnătură, iar clasele Patrulater\_1 și Patrulater\_2 implementează fiecare în mod diferit cele două metode;
- c) **programul este corect și după rulare va afișa 100.0 40.0 200.0 60.0;**
- d) programul este incorect deoarece în funcția main se atribuie instanței d a interfeței Dreptunghi un obiect din clasa Patrulater\_1, iar instanței p a interfeței Patrat un obiect de tip Patrulater\_2 (care, de fapt, abstractizează noțiunea de dreptunghi).

**17.** Considerăm următorul program Java:

```

class Cls
{
    int a,b;
    public Cls(int x, int y) { a=x; b=y; f(); g(); }
    void f()
    {
        while(a<b)
        {
            b=b-2*a;
            a=a+b/10;
        }
    }
    void g() { System.out.println(a+" "+b); }
}

public class Test
{
    public static void main(String[] args)
    {
        Cls ob = new Cls(5,100);
    }
}

```

}

După executarea programului, pe ecran se va afișa:

- a) 20 -22
- b) 22 22**
- c) 35 10
- d) 5 100

**18.** Considerăm următorul program Java:

```
class C
{
    int a,b;

    public C(int x, int y){a=x; b=y;}

    void f()
    {
        if(a<b) { a++; b--; g(); }
    }

    void g()
    {
        if(b>=a) { a++; b--; f(); }
    }

    void afisare() { System.out.println(a+" "+b);}
}

public class teste_grila
{
    public static void main(String[] args)
    {
        C ob = new C(2,10);
        ob.f(); ob.g();
        ob.afisare();
    }
}
```

După executarea programului, pe ecran se va afișa:

- a) 57
- b) 6 6
- c) 2 10
- d) 7 5**

**19.** Considerăm următorul program Java:

```
class C
{
    static int x = 0;
    static int f() { return (++x)*(x--); }
}

public class teste_grila
```

```

{
    public static void main(String[] args)
    {
        System.out.println(C.f()+" "+C.f()+" "+C.f());
    }
}

```

După executarea programului, pe ecran se va afișa:

- a) 1 1 1
- b) 1 2 3
- c) 1 2 6
- d) 0 0 0

**20.** Considerăm următorul program Java:

```

class C
{
    static int x=0;
    static void f()
    {
        x = (++x)*(x--);
        System.out.print(x+" ");
    }
}

public class teste_grila
{
    public static void main(String[] args)
    {
        C.f();C.f();C.f();
    }
}

```

După executarea programului, pe ecran se va afișa:

- a) 0 0 0
- b) 1 4 25
- c) 1 -1 1
- d) 2 4 16

**21.** Un fir de execuție poate intra în starea "blocat" (blocked) astfel:

- a) prin apelul metodei *sleep()*;
- b) automat de către sistemul de operare;
- c) prin apelul metodei *block()*;
- d) prin apelul metodei *wait()*.

**22.** Prin modalitatea sa de tratare a excepțiilor, Java oferă următoarele avantaje față de mecanismul tradițional de tratare a erorilor:

- a) există o metodă care se ocupă de acest lucru;
- b) separarea codului pentru tratarea unei erori de codul în care ea poate să apară;
- c) propagarea unei erori până la un analizor de excepții corespunzător;

d) gruparea erorilor dupa tipul lor.

23. O subclasă a unei clase abstracte poate fi instanțiată numai dacă:

- a) se folosește cuvântul cheie `abstract`;
- b) suprascrise fiecare metodă declarată abstractă în superclasa sa și furnizează implementări pentru toate acestea;
- c) se folosește moștenirea multiplă;
- d) subclasă abstractă nu poate fi instanțiată.

24. Care este rolul declarațiilor `import`?

- a) Permite referirea claselor fără utilizarea de prefixe;
- b) Permite importul imaginilor folosite;
- c) Elimină necesitatea declarării variabilelor;
- d) Elimină apelurile directe ale funcțiilor fără clase.

25. Indicați pe care dintre sistemele de operare următoare pot fi rulate aplicațiile Java:

- a) Windows
- b) UNIX
- c) Mac OS X
- d) Linux

26. Care dintre liniile de mai jos realizează atribuirea șirului „Hello Java” variabilei `s` ?

- a) `String s = "Hello Java";`
- b) `String s[] = "Hello Java";`
- c) `new String s = "Hello Java";`
- d) `String s = new String("Hello Java");`

27. Ce se afișează dacă se execută următorul cod Java:

```
String s = new String( "Computer" );
if( s == "Computer" )
    System.out.println( "Equal A" );
if( s.equals( "Computer" ) )
    System.out.println( "Equal B" );
```

- a) Eroare la compilare, deoarece operatorul `==` nu se poate aplica pentru tipul `String`
- b) Se afișează mesajul “Equal A”
- c) Se afișează mesajul “Equal B”
- d) Se afișează ambele mesaje, “Equal A”, respectiv “Equal B”

28. Pentru `size=4`, forma tabloului `triArray` este

```
int[][][] makeArray( int size) {
    int[][][] triArray = new int[size] [];
    int val=1;
```



```

        for( int i = 0; i < triArray.length; i++ ){
            triArray[i] = new int[i+1];
            for( int j=0; j < triArray[i].length; j++ )
                {
                    triArray[i][j] = val++;
                }
        }
    return triArray;
}

```

a)

```

1 2 3 4
5 6 7
8 9
10

```

b)

```

1 4 9 16

```

c)

```

1 2 3 4

```

d)

```

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

```

e)

```

1
2 3
4 5 6
7 8 9 10

```

**29.** Dacă tabloul `arr[]` conține doar valori pozitive, ce va returna următoare funcție?

```

public int guessWhat( int arr[] ){
    int x= 0;
    for( int i = 0; i < arr.length; i++ )
        x = x < arr[i] ? arr[i] : x;
    return x;
}

```

- a) Returnează indexul celui mai mare element din tablou
- b) Returnează true/false dacă există un element care se repetă
- c) Returnează elementul cu valoarea cea mai mare
- d) Returnează numărul elementelor din tablou

**30.** La execuția programului Java

```

class T1 {

```

```
public static void main(String[] a){
    int x = 5;
    int y = (x = 4)*x; System.out.println(y);
}
}
```

se va afișa:

- a) 20
- b) 16**
- c) 5
- d) 4

**31.** În Java, metoda `clone()` a clasei `Object`

- a) Creeaza un obiect nou al clasei folosind constructorul implicit
- b) Creeaza și returnează o copie a obiectului curent**
- c) Returneaza codul asociat constructorului implicit
- d) Testează dacă obiectul specificat este o clonă a obiectului current

**32.** Un fir de executare Java este

- a) O instanță a unei clase derivate din clasa Thread**
- b) O instanță a unei clase care implementează interfața Runnable**
- c) Fie un obiect al unei clase a carei superclasă este clasa Thread, fie un obiect al unei clase care implementează interfața Runnable**

**33.** Fie următorul program Java:

```
public class Asignare {
    public static void main(String args[]){
        int a = 3;int b = (a = 2) * a;int c = b * (b = 5) ;
        System.out.println("a = " + a + ", b = " + b + ", c = " + c);}}
}
```

Ce va afișa acesta la execuție?

- a) a = 2, b = 4, c = 20
- b) a = 2, b = 5, c = 20**
- c) a = 2, b = 5, c = 25
- d) a = 3, b = 6, c = 30

**34.** Urmatorul subprogram Java:

```
int as=3, bs=2, cs=4;
System.out.print(((as < bs++) & (cs++ < bs)) + " ");
System.out.println(as + " " + bs + " " + cs);
System.out.print(((as < bs++) && (os++ < bs++)) + " ");
System.out.println(as + " " + bs + " " + cs);
```

Afișează :

- a) Eroare la compilare: nu se poate aduna o valoare booleana cu un String;
- b) Subprogramul se compilează și la execuție afișează :false 3 3 5 false 3 4 4

- c) Subprogramul se compileaza și la execuție afisează: false 3 3 5 false 3 4 5  
 e) Subprogramul se compileaza și la execuție afisează false 3 3 5 false 3 5 6

**35.** Considerăm următorul program Java:

```
public class test
{
    public static void main(String args[])
    {
        int v[ ]={-2,4,-5,-6,0,2}, suma=0,i;
        for(i=0;i<5;i++)
            if(v[i]<-2) suma+=v[i];
        System.out.println(suma);
    }
}
```

După executarea programului, pe ecran se va afișa:

- a) -7  
 b) 0  
 c) -11  
 d) -13

**36.** Secvența uramatoare:

```
public class test {
public static void main(String args[]){
    String sir = "Programare in Java, C++ este usoara" ;
    String atom[]= sir.split(" ") ;
    System.out.println(atom.length) ;
}}

```

Afisează:

- a) Eroare la compliare pentru ca nu se specifica numarul de elemente ale tabloului atom  
 b) 6  
 c) 7  
 d) 8

**37.** Secvența uramatoare:

```
public class test{
public static void main(String args[]){
    String sir = "Programare in Java, C++ este usoara" ;
    String atom[]= sir.split(" ") ;
    System.out.println(atom[0].length()) ;
}}

```

Afisează:

- a) Eroare la compliare pentru ca nu se specifica numarul de elemente ale tabloului atom  
 b) 6  
 c) 10  
 d) 9

**38.** Ce se va afișa la execuția următorului program Java?

```
interface I1{
    float x=2.3f;
}
public class Test implements I1{
    public static void main(String [] args){
        System.out.print(x+" ");
        x=6.7f;
        System.out.print(x);
    }
}
```

a) Va apărea eroare la compilare deoarece valoarea variabilei `x` nu se mai poate modifica;

b) La execuție se va afișa: 2.3f 6.7f;

c) La execuție se va afișa: 2.3f 2.3f;

d) La execuție se va afișa: 2.3 6.7;

**39.** Următorul program Java:

```
class C1{
    int x=1;
    void f(int x){
        this.x=x;}
    int getX_C1(){
        return x;}}
class C2 extends C1{
    float x=5.0f;
    int f(int x){
        super.f((int)x);}
    float getX_C2(){
        return x;}}
public class Test{
    public static void main(String []args){
        C2 obiect = new C2();
        obiect.f(4);
        System.out.print(obiect.getX_C2() + " ");
        System.out.println(obiect.getX_C1());}}
```

Afișează:

a) Programul este corect și va afișa la execuție 5 4;

b) Programul este corect și va afișa la execuție 4.0 4;

c) Va apărea eroare la compilare deoarece în clasa C2 s-a suprascris gresit atributul `x` din clasa C1;

d) Va apărea eroare la compilare deoarece metoda suprascrisă `f()` din clasa C2 întoarce un tip diferit de `void`;

**40.** O subclasă a unei clase abstracte poate fi instanțiată numai dacă:

a) Se folosește cuvântul cheie `abstract`;

b) Suprascrie fiecare metodă declarată abstractă în superclasa sa, și furnizează implementări pentru toate acestea;

c) Se folosește moștenirea multiplă;

d) O subclasă abstractă nu poate fi instanțiată;

**41.** Urmatorul program Java:

```
class C1{
    int x=1;
    void f(int x){
        this.x=x;
    }
    int getX_C1(){
        return x;
    }
}
class C2 extends C1{
    float x=5.0f;
    void f(int x){
        super.f((int)x);
    }
    float getX_C2(){
        return x;
    }
}
public class Test{
    public static void main(String []args){
        C2 obiect = new C2();
        obiect.f(4);
        System.out.print(obiect.getX_C2() + " ");
        System.out.println(obiect.getX_C1());
    }
}
```

Afișează:

- a) Programul este corect și va afișa la execuție 5.0 4;
- b) Programul este corect și va afișa la execuție 4.0 4;
- c) Va apărea eroare la compilare deoarece în clasa C2 s-a suprascries gresit atributul x din clasa C1;
- d) Programul este corect și va afișa la execuție 5.0 5;

**42.** Tipurile referința în Java sunt:

- a) tabloul, clasa, interfața
- b) clasa, interfata
- c) int, flout, double, char, String
- d) String si null

**43.** Secvența următoare:

```
public class numar_43_nou {
    public static void main(String args[])
    {
        String sir="Examen";
        sir.toUpperCase();
        System.out.println(sir);
    }
}
```

Afișează:

- a) EXAMEN

- b) Examen
- c) eXamen
- d) **Examen**

44. Secvența urătoare:

```
public class numar_44_nou {
    public static void main(String args[])
    {
        String sir1="Programare in Java";
        String sir2 =sir1.substring(4,8);
        System.out.println(sir2.toUpperCase());    }
    }
```

Afișează:

- a) ogramare
- b) rama
- c) **RAMA**
- d) Java

45. Secvența urătoare:

```
public class Test {
    public static void main(String args[])
    {
        int numar = 1;
        for (int x = 0; x < 4; x++)
            numar = numar << 1;
        System.out.println(numar); }
    }
```

Afișează:

- a) **16**
- b) 32
- c) 8
- d) 3

46. O clasă abstractă în Java :

- a) conține doar metode abstracte
- b) **conține metode abstarcte, dar poate conține și metode definite**
- c) poate fi instanțiată
- d) **nu poate fi instanțiată, dar se pot defini referințe către acesta**

47. Secveța uramatoare:

```
public class test {
    public static void main(String args[]){
```

```
String sir = "Programare in Java, C++ este usoara" ;
String atom[]= sir.split("[ ,]") ;
System.out.println(atom.length) ;
}}
```

Afisează:

- a) Eroare la compiliere pentru ca nu se specifica numarul de elemente ale tabloului atom
- b) 2**
- c) 6
- d) 5

**48.** O clasă declarată `final`

- a) clasa nu poate fi instanțiată
- b) orice cod exterior are acces la codul clasei
- c) implementează o interfață
- d) nu poate avea subclase**

**49.** Compoziția reprezintă

- a. O relație de tip IS-A
- b. O relație de tip Can Do
- c. orelație de tip HAS-A.**
- d. Niciorelație

**50.** Se consideră următorul fragment de cod:

```
for(int i=0;i<2;i++) {
    for(int j=0;j<3;j++) {
        if(i==j) {
            continue;
        }
        System.out.println("i="+i+" j="+j);
    }
}
```

Care linii vor face parte din output?

- a) i=0 j=0
- b) i=0 j=1**
- c) i=0 j=2**
- d) i=1 j=0**
- e) i=1 j=1

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Algoritmi si structuri de date**



1. Se dă următorul algoritm:

```

for i = 1, n
    poz[i] = 1
endfor
for i = 1, n-1
    for j = i+1, n
        if x[j] < x[i] then poz[i] = poz[i] + 1
                        else poz[j] = poz[j] + 1
        endif
    endfor
endfor

```

Știind că datele de intrare sunt  $n = 7$  și vectorul  $x = (9, 15, 23, 2, 5, 4, 8)$  care vor fi valorile vectorului  $poz$  la sfârșitul algoritmului?

- a. (5, 6, 7, 1, 2, 3, 4)
- b. (5, 6, 7, 1, 3, 2, 4)**
- c. (6, 5, 7, 1, 2, 3, 4)
- d. (1, 2, 3, 4, 5, 6, 7)

2. Se dă următoarea funcție recursivă

- 1) int inaltime(NodArb \*rad)
- 2) // returneaza inaltimea unui arbore binar
- 3) {
- 4)     if(rad == NULL) return 0;
- 5)     .....
- 6)     return 1 + max(inaltime(rad->stang), inaltime(rad->drept));
- 7)
- 8) }

Ce instrucțiuni trebuie scrise în linia de cod 5) pentru ca funcția să returneze înălțimea unui arbore binar?

- a. instrucțiunea vidă
- b. int inaltime =0;
- c. else if(rad->stang == NULL && rad->drept == NULL) return 0;**
- d. else

3. Se dă următoarea funcție

```

int cautare(int x[], int first, int last, int value)
{
    int mid;
    if(first > last) return -1;
    mid = (first + last) / 2;
    if (x[mid] == value) return mid;
    if(x[mid] < value) return cautare(x, mid + 1, last, value);
    else return cautare(x, first, mid - 1, value);
}

```

Dacă vectorul  $x = (2, 4, 5, 8, 9, 15, 23)$ , care va fi rezultatul apelării funcției  $cautare(x, 2, 6, 8)$  ?

- a. -1
- b. 2
- c. 3**
- d. 1

4. Parcurgerea în preordine a arborelui binar din Fig. 1 va afișa

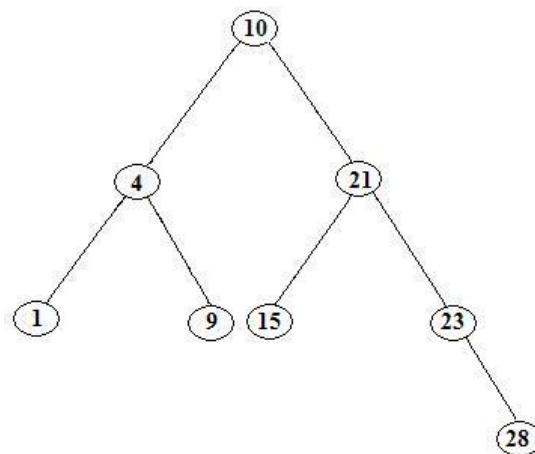


Fig. 1

- 10, 4, 1, 9, 21, 15, 28, 23
  - 10, 4, 1, 9, 23, 21, 28, 15
  - 1, 4, 9, 10, 15, 21, 23, 28
  - 10, 4, 1, 9, 21, 15, 23, 28
5. Parcurgerea în inordine a arborelui binar din Fig. 1 va afișa
- 10, 4, 1, 9, 21, 15, 23, 28
  - 1, 4, 9, 10, 15, 21, 23, 28
  - 1, 4, 9, 10, 15, 21, 28, 23
  - 1, 4, 9, 10, 21, 23, 28, 15
6. Parcurgerea în postordine a arborelui binar din Fig. 1 va afișa
- 1, 4, 9, 10, 15, 21, 23, 28
  - 1, 4, 9, 10, 15, 21, 28, 23
  - 1, 9, 4, 15, 28, 23, 21, 10
  - 1, 9, 4, 15, 23, 28, 21, 10
7. Ce returnează următoarea funcție dacă rad este pointer la rădăcina unui arbore binar nenul?
- ```

int fct(NodArb *rad)
{
    if(rad == NULL) return 0;
    return 1 + fct(rad->stang) + fct(rad->drept);
}
  
```
- 0
  - 1
  - numărul de noduri terminale ale arborelui
  - numărul de noduri ale arborelui.
8. Ordinul de complexitate a algoritmului Bubblesort (metoda bulelor) este
- $O(n)$
  - $O(n^2)$
  - $O(n \log n)$
  - $O(n^3)$

9. Cel mai rău caz pentru algoritmul de sortare rapidă este cazul în care
  - a. **vectorul este deja sortat**
  - b. vectorul nu este creat aleator
  - c. toate elementele vectorului sunt pare
  - d. vectorul conține și elemente negative
10. Câte comparații se fac dacă se folosește algoritmul de căutare secvențială pentru căutarea elementului 12 în vectorul (2, 3, 6, 9, 10, 25)?
  - a. **6**
  - b. 5
  - c. 3
  - d. 1
11. Câte comparații se fac dacă se folosește algoritmul de căutare binară pentru căutarea elementului 12 în vectorul (2, 3, 6, 9, 10, 25)?
  - a. 6
  - b. 5
  - c. **3**
  - d. 1
12. O listă liniară în care inserările în lista se fac pe la un capăt, iar ștergerile pe la celălalt capăt se numește
  - a. stivă
  - b. vector
  - c. **coadă**
  - d. arbore
13. Care din următorii algoritmi au ordinul de complexitate  $O(n \log n)$ ?
  - a. Bubblesort (sortarea cu metoda bulelor)
  - b. **Mergesort (sortarea prin interclasare)**
  - c. sortarea prin inserare
  - d. **Quicksort(sortarea rapidă).**
14. Cel mai rău caz pentru algoritmul de căutare secvențială este cazul în care
  - a. elementul căutat este la mijlocul listei
  - b. **elementul căutat nu se află în listă**
  - c. **elementul căutat este pe ultima poziție în listă**
  - d. vectorul este ordonat crescator
15. Timpul de execuție al unui algoritm se măsoară în
  - a. numărul de kilocteți necesari
  - b. numărul de instrucțiuni ale algoritmului
  - c. **numărul de operații cheie**
  - d. numărul de milisekunde necesar executării.
16. Ordinul de complexitate a algoritmului de căutare binară este
  - a.  $O(n)$
  - b.  $O(n^2)$
  - c.  $O(n \log n)$
  - d.  **$O(\log n)$**
17. O problemă se poate rezolva prin trei algoritmi, unul cu ordinul de complexitate  $O(n)$ , altul cu ordinul  $O(\log n)$  și al treilea cu ordinul  $O(n \log n)$ . Care este cel mai bun?

- a. cel cu ordinul  $O(n)$
- b. cel cu ordinul  $O(\log n)$
- c. cel cu ordinul  $O(n \log n)$
- d. Toate sunt la fel.

18. Se dă următorul algoritm:

```

for i = 1, n - 1
    index_min = i
    for j = i + 1, n
        if x[index_min] > x[j] then index_min = j
    endif
    endfor
    if i ≠ index_min then
        temp = x[i]
        x[i] = x[index_min]
        x[index_min] = temp
    endif
endfor

```

Care vor fi valorile vectorului x după terminarea pasului  $i = 3$  știind că la intrare avem valorile  $n = 7$  și vectorul  $x = (9, 15, 23, 2, 5, 4, 8)$ ?

- a. (2, 4, 5, 9, 15, 23, 8)
- b. (2, 5, 9, 15, 23, 4, 8)
- c. (2, 5, 9, 15, 4, 23, 8)
- d. (2, 4, 5, 9, 23, 15, 8)

19. Se dă următorul algoritm. Care vor fi valorile vectorului x după terminarea pasului  $i = 5$ , știind că la intrare avem valorile  $n = 7$  și  $x = (9, 15, 23, 2, 5, 4, 8)$ ?

```

for i = 2, n
    elem = x[i]
    j = i - 1
    while j >= 1 and x[j] > x[i]
        j = j - 1
    endwhile
    pozitie = j + 1
    for j = i, pozitie + 1, -1
        x[j] = x[j - 1]
    endfor
    x[pozitie] = elem
endfor

```

- a. (2, 4, 5, 9, 15, 23, 8)
- b. (2, 5, 9, 15, 23, 4, 8)
- c. (2, 5, 9, 15, 4, 23, 8)
- d. (2, 4, 5, 8, 9, 15, 23)

20. Se consideră următoarea secvență de operații într-o stivă: push(2), push(10), push(8), pop(), push(9), push(6), pop(), pop(), push(7), push(3), pop(), pop(), pop(), pop(). În ce ordine vor fi scoase din stivă elementele? (push = inserare, pop = ștergere)
- (2, 10, 8, 9, 6, 7, 3)
  - (3, 7, 6, 9, 8, 10, 2)
  - (8, 6, 9, 3, 7, 10, 2)**
  - (6, 9, 3, 7, 8, 10, 2)
21. Se consideră următoarea secvență de operații într-o coadă: enqueue(2), enqueue(10), enqueue(8), dequeue(), enqueue(9), enqueue(6), dequeue(), dequeue(), enqueue(7), enqueue(3), dequeue(), dequeue(), dequeue(). În ce ordine vor fi scoase din coadă elementele? (enqueue = inserare, dequeue = ștergere)
- (2, 10, 8, 9, 6, 7, 3)**
  - (3, 7, 6, 9, 8, 10, 2)
  - (8, 6, 9, 3, 7, 10, 2)
  - (6, 9, 3, 7, 8, 10, 2)
22. Se consideră următoarea funcție care caută o valoare dată într-o listă înlănțuită. val este variabila a cărei valoare este căutată, iar head este un pointer la capul listei în care se face căutarea.
- 1) NOD \*cauta(NOD \*head, int val)
  - 2) {
  - 3)   NOD \*iter = head; int gasit=0;
  - 4)   while (.....)
  - 5)   {
  - 6)     if (iter -> info == val) gasit = 1;
  - 7)     else iter = iter -> link;
  - 8)   }
  - 9)   if(gasit) return iter;
  - 10) else return NULL;
  - 11) }

Cum trebuie completată linia de cod 4 astfel încât funcția să ruleze corect și să returneze un pointer la nodul cu valoarea căutată sau NULL dacă valoarea nu a fost găsită în listă?

- !gasit && iter != NULL**
  - !gasit
  - iter!=NULL && !gasit**
  - gasit ==0
23. Se consideră următoarea funcție care are drept variabilă de intrare un pointer la capul unei liste înlănțuite. Ce face această funcție?
- 1) NOD \*fct(NOD \*head)
  - 2) {
  - 3)   if (head == NULL) return NULL;
  - 4)   head = head -> link;
  - 5)   return head;
  - 6) }
  - a. returnează NULL
  - b. returnează un pointer la capul listei

- c. elimină primul nod al listei și returnează un pointer la noul cap al listei
- d. returnează NULL dacă lista este vidă

24. Cel mai rău caz pentru algoritmul de sortare prin selecție este cazul în care

- a. vectorul este ordonat descrescător
- b. cel mai mare element al vectorului se află în prima poziție în vector
- c. nu există un cel mai rău caz
- d. vectorul este ordonat crescător

25. Cel mai bun caz pentru algoritmul de sortare prin metoda bulelor (Bubblesort) este cazul în care

- a. cel mai mic element al vectorului se află pe prima poziție în vector
- b. cel mai mare element al vectorului se află în ultima poziție în vector
- c. nu există un cel mai bun caz
- d. vectorul este ordonat crescător

26. Se consideră lista înlănțuită cu elemente numere întregi din Fig. 2:

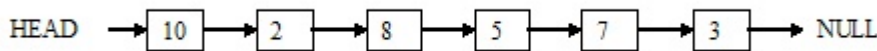


Fig. 2

Data următoarea definiție a tipului de date ce corespunde unui nod al listei,

```

struct NOD
{
    int info;
    NOD *link;
};
  
```

ce va afișa următoarea funcție, dacă este apelată prin print(HEAD)?

```

void print(NOD *head)
{
    NOD *iter=head;
    while(iter!=NULL)
    {
        cout << iter->info<<" ";
        iter=iter->link;
    }
}
  
```

- a. 10, 2, 8, 5, 7, 3,
- b. 3, 7, 5, 8, 2, 10,
- c. 2, 8, 5, 7, 3,
- d. 2, 8, 5, 7,

27. Se consideră lista înlănțuită cu elemente numere întregi din Fig. 2. Data următoarea definiție a tipului de date ce corespunde unui nod al listei,

```

struct NOD
{
    int info;
    NOD *link;
};
  
```

ce va afișa următoarea funcție, dacă este apelată prin print(HEAD)?

```
void print(NOD *head)
{
    NOD *iter=head;
    while(iter->link !=NULL)
    {
        cout << iter->info<<" ";
        iter=iter->link;
    }
}
```

- a. 10, 2, 8, 5, 7, 3,
- b. 10, 2, 8, 5, 7,**
- c. 2, 8, 5, 7, 3, 10,
- d. 2, 8, 5, 7,

28. Se consideră lista înlănțuită cu elemente numere întregi din Fig. 2. Dată următoarea definiție a tipului de date ce corespunde unui nod al listei,

```
struct NOD
{
    int info;
    NOD *link;
};
```

ce va afișa următoarea funcție, dacă este apelată prin print(HEAD)?

```
void print(NOD *head)
{
    NOD *iter=head;
    while(iter->link !=NULL)
    {
        iter=iter->link;
        if ((iter-> info)%2) cout << iter->info<<" ";
    }
}
```

- a. 10, 2, 8, 5, 7, 3,
- b. 10, 3, 7, 5, 8, 2,
- c. 2, 8, 5, 7, 3,
- d. 5, 7, 3,**

29. Se dă următoarea funcție

```
void interclasare(int x[], int prim, int mijloc, int ultim, int C[])
{
    // lista A: x[prim ..mijloc]
    // lista B: x[mijloc+1 ..ultim]
    // lista C: C[0.. ultim - prim]
    int iterA = prim, iterB = mijloc+1, iterC = 0;
    while (iterA <= mijloc && iterB <=ultim)
        if (x[iterA] < x[iterB])
            C[iterC ++]= x[iterA ++];
        else C[iterC ++]= x[iterB ++];
    while (iterA <= mijloc)
        C[iterC ++]= x[iterA ++];
}
```

```

while (iterB <= ultim)
    C[iterC ++]= x[iterB ++];
}

```

Dacă vectorul  $x = (9, 15, 23, 25, 4, 5, 8)$ , care va fi vectorul  $C$  la apelarea funcției interclasare ( $x, 0, 3, 6, C$ )?

- a. 4, 5, 8, 9, 15, 23, 25
- b. 0, 0, 0, 0, 0, 0, 0
- c. 9, 4, 15, 5, 23, 8, 25
- d. 9, 15, 23, 25, 4, 5, 8

30. Se dă următoarea funcție, care apelează funcția interclasare descrisă la exercițiul 29.

```

void mergesort(int x[], int prim, int ultim)
{
    if (prim < ultim)
    {
        int mijloc = (prim + ultim)/2;
        mergesort(x, prim, mijloc);
        mergesort(x, mijloc + 1, ultim);
        int C[ultim - prim + 1];
        interclasare(x, prim, mijloc, ultim, C);
        for (int i = prim; i <= ultim; i++)
            x[i]=C[i-prim];
    }
}

```

La apelul funcției `mergesort(x, 0, 6)` unde vectorul  $x = (9, 15, 23, 2, 4, 5, 8)$ , de câte ori va fi apelată funcția `mergesort` (incluzând apelul inițial)?

- a. 1
- b. 3
- c. 13
- d. 7

31. Ordinul de complexitate a algoritmului de sortare prin inserare este

- a.  $O(n)$
- b.  $O(n^2)$
- c.  $O(n \log n)$
- d.  $O(n^3)$

32. Ordinul de complexitate a algoritmului de căutare secvențială este

- a.  $O(n)$
- b.  $O(n^2)$
- c.  $O(n \log n)$
- d.  $O(\log n)$

33. O listă liniară în care inserările și ștergerile în lista se fac pe la un singur capăt se numește

- a. stivă
- b. vector
- c. coadă
- d. arbore

34. Ordinul de complexitate a algoritmului de sortare prin selecție este



- a.  $O(n)$
- b.  $O(n^2)$
- c.  $O(n \log n)$
- d.  $O(n^3)$

35. Se dă următoarea funcție în care front și rear sunt variabile globale și reprezintă pointeri la nodurile unei liste liniare reprezentate simplu înălțuit, front la primul nod al listei, iar rear pointer la ultimul nod.

```
void fct(int a)
{
    nod *p = new nod;
    if (p != NULL)
    {
        p->info = a;
        p->link = NULL;
        if(rear!=NULL) rear->link=p;
        else front=p;
        rear = p;
    }
    else cout << "OVERFLOW"<<endl;
}
```

Ce face această funcție?

- a. Inserează un nod la începutul listei.
- b. Inserează un nod la sfârșitul listei.
- c. Șterge nodul de la începutul listei.
- d. Șterge nodul de la sfârșitul listei.

36. Cel mai rău caz pentru algoritmul de sortare prin inserare este cazul în care

- a. vectorul este deja sortat
- b. vectorul este crescător și se dorește sortarea lui în ordine descrescătoare
- c. vectorul este descrescător și se dorește sortarea lui în ordine crescătoare
- d. vectorul conține și elemente negative

37. Cel mai bun caz pentru algoritmul de sortare prin inserare este cazul în care

- a. vectorul este deja sortat în ordinea dorită
- b. vectorul nu este creat aleator
- c. vectorul este descrescător și se dorește sortarea lui în ordine crescătoare
- d. vectorul conține și elemente negative

38. Numărul minim de comparații între elementele unui vector cu n elemente care este sortat cu algoritmul de sortare prin inserare este

- a. n
- b. n+1
- c. n-1
- d.  $\log n$ .

39. Numărul minim de comparații între elementele unui vector cu n elemente care este sortat cu algoritmul de sortare prin metoda bulelor (Bubblesort) este

- a. n-1
- b. n+1
- c. n

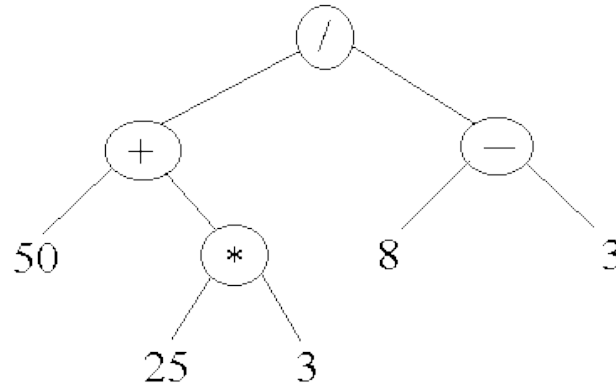
- d.  $n \log n$ .
40. Numărul maxim de comparații între elementele unui vector cu  $n$  elemente care este sortat cu algoritmul de sortare prin metoda bulelor (Bubblesort) este
- $n!$
  - $n(n+1)/2$
  - $n(n-1)/2$
  - $n \log n$ .
41. Câte comparații se fac dacă se folosește algoritmul de căutare secvențială pentru căutarea elementului 9 în vectorul (8, 3, 5, 9, 11, 2)?
- 6
  - 5
  - 3
  - 4
42. Câte comparații se fac dacă se folosește algoritmul de căutare binară pentru căutarea elementului 19 în vectorul (1, 2, 3, 5, 8, 9, 19)?
- 6
  - 5
  - 3
  - $O(\log 7)$
43. Dacă se aplicăm metoda bulelor (bubblesort) pentru sortarea vectorului  $x = (9, 15, 23, 25, 4, 8, 5)$ , cum se va modifica vectorul  $x$  după prima parcurgere a sa?
- 4, 5, 8, 9, 15, 23, 25
  - 9, 15, 23, 4, 8, 5, 25
  - 9, 4, 15, 5, 23, 8, 25
  - 9, 15, 23, 25, 4, 5, 8
44. Dacă se aplicăm metoda bulelor (bubblesort) pentru sortarea vectorului  $x = (9, 15, 23, 25, 4, 8, 5)$ , cum se va modifica vectorul  $x$  după două parcurgeri ale sale?
- 4, 5, 8, 9, 15, 23, 25
  - 9, 15, 4, 8, 5, 23, 25
  - 9, 4, 15, 5, 23, 8, 25
  - 9, 15, 23, 25, 4, 5, 8
45. Dacă se aplicăm sortarea prin inserare pentru sortarea vectorului  $x = (9, 15, 23, 25, 4, 8, 5)$ , care este primul element al vectorului a cărui analiză va implica efectuarea de modificări asupra vectorului?
- 15
  - 23
  - 25
  - 4
46. Care din următoarele afirmații sunt adevărate?
- La aplicarea algoritmului de sortare rapidă elementul din mijloc este mutat pe prima poziție.
  - La aplicarea algoritmului de sortare rapidă elementul de pe prima poziție este mutat pe poziția din mijloc.
  - La aplicarea algoritmului de sortare rapidă se alege un element din listă, numit pivot și se rearanjează lista, prin interschimbări, inclusiv prin mutarea pivotului pe o altă poziție, astfel încât toate elementele mai mici decât pivotul să fie poziționate înaintea lui, iar toate elementele mai mari să fie poziționate după acesta.

d. La aplicarea algoritmului de sortare rapidă nu se alege niciun element pivot.

47. Care din următoarele afirmații sunt adevărate?

- a. Arborele din figura Fig. 1 este un arbore binar.
- b. Arborele din figura Fig. 1 nu este un arbore binar.
- c. Arborele din figura Fig. 1 este un arbore binar de căutare.
- d. Arborele din figura Fig. 1 nu este un arbore binar de căutare.

48. Parcurgerea în preordine a arborelui din Fig. 3 va afișa



**Fig. 3**

- a. /, +, 50, \*, 25, 3, 8, -, 3
- b. /, 50, +, \*, 3, 25, 8, -, 3
- c. 50, +, 25, \*, 3, 8, -, 3, /
- d. /, +, 50, \*, 25, 3, -, 8, 3

49. Parcurgerea în inordine a arborelui din Fig. 3 va afișa

- a. /, +, 50, \*, 25, 3, 8, -, 3
- b. 50, +, 25, \*, 3, /, 8, -, 3
- c. 50, +, 25, \*, 3, 8, -, 3, /
- d. 50, /, +, \*, 25, 3, -, 8, 3

50. Parcurgerea în postordine a arborelui din Fig. 3 va afișa

- a. 50, 25, 3, \*, +, 8, 3, -, /
- b. /, 50, +, \*, 3, 25, 8, -, 3
- c. 50, +, 25, \*, 3, 8, -, 3, /
- d. /, +, 50, \*, 25, 3, -, 8, 3

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Tehnici avansate de programare**

1. Complexitatea minimă a unui algoritm care calculează numărul tuturor submulțimilor unei mulțimi cu  $n$  elemente este:
  - a)  $\mathcal{O}(n^2)$
  - b)  $\mathcal{O}(2^n)$
  - c)  $\mathcal{O}(n)$
  - d)  $\mathcal{O}(n!)$
2. Complexitatea minimă a unui algoritm care afișează toate submulțimile unei mulțimi cu  $n$  elemente este:
  - a)  $\mathcal{O}(n^2)$
  - b)  $\mathcal{O}(2^n)$
  - c)  $\mathcal{O}(n)$
  - d)  $\mathcal{O}(n!)$
3. Complexitatea minimă a unui algoritm care calculează numărul modurilor în care pot fi așezate  $n$  cărți pe un raft suficient de lung este:
  - a)  $\mathcal{O}(n^2)$
  - b)  $\mathcal{O}(2^n)$
  - c)  $\mathcal{O}(n)$
  - d)  $\mathcal{O}(n!)$
4. Complexitatea minimă a unui algoritm care afișează toate modurile în care pot fi așezate  $n$  cărți pe un raft suficient de lung este:
  - a)  $\mathcal{O}(n^2)$
  - b)  $\mathcal{O}(2^n)$
  - c)  $\mathcal{O}(n)$
  - d)  $\mathcal{O}(n!)$
5. Considerăm următorul program în limbajul C:

```
#include<stdio.h>
int main()
{
    int i,j,n,a[101];
    scanf("%d",&n);
    for(i=0;i<n;i++) scanf("%d",&a[i]);
    i=0;
    while((i<n)&&(a[i]<0)) i++;
    j=n-1;
    while((j>=0)&&(a[j]>=0)) j--;
    if(i>=j) printf("1");
    else printf("0");
    return 0;
}
```

Complexitatea algoritmului implementat în acest program este:

- a)  $\mathcal{O}(n^2)$
- b)  $\mathcal{O}(\log_2 n)$
- c)  $\mathcal{O}(n^3)$
- d)  $\mathcal{O}(n)$

6. Se consideră un șir format din maxim 100 de numere naturale distincte cuprinse între 1 și 1000. Complexitatea minimă a unui algoritm care să afișeze numerele din șir în ordine crescătoare este:

- a)  $\mathcal{O}(n^2)$
- b)  $\mathcal{O}(n^3)$
- c)  $\mathcal{O}(n \log_2 n)$
- d)  $\mathcal{O}(n)$

7. Considerăm următorul program în limbajul C:

```
#include<stdio.h>
int main()
{
    int a[100],i,j,n,s;
    scanf("%d",&n);
    for(i=0;i<n;i++) scanf("%d",&a[i]);

    i = s = 0;
    while(i<n)
    {
        j=i+1;
        while((j<=n) && (a[i]==a[j])) j++;
        s++;
        i=j;
    }
    printf("\n\n%d\n", s);
    return 0;
}
```

Complexitatea algoritmului implementat în acest program este:

- a)  $\mathcal{O}(n^2)$
- b)  $\mathcal{O}(\log_2 n)$
- c)  $\mathcal{O}(n^3)$
- d)  $\mathcal{O}(n)$

8. Considerăm următorul program în limbajul C:

```
#include<stdio.h>
int main()
{
    int i,j,n,a[101];
```

```

scanf("%d", &n);
for(i=0; i<n; i++) scanf("%d", &a[i]);
i=0;
while((i<n) && (a[i]<0)) i++;
j=n-1;
while((j>=0) && (a[j]>=0)) j--;
if(i>=j) printf("1");
else printf("0");
return 0;
}

```

Programul afișează:

- a) 1 dacă și numai dacă tabloul  $a$  este sortat crescător și 0 altfel;
- b) 1 dacă și numai dacă toate valorile din tabloul  $a$  sunt pozitive și 0 altfel;
- c) 1 dacă și numai dacă în tabloul  $a$  valorile negative se află înaintea celor pozitive și 0 altfel;
- d) 1 dacă și numai dacă toate valorile din tabloul  $a$  sunt strict negative și 0 altfel.

9. Considerăm următorul program în limbajul C:

```

#include<stdio.h>
int main()
{
    int a[100], i, j, n, s;
    scanf("%d", &n);
    for(i=0; i<n; i++) scanf("%d", &a[i]);

    i = s = 0;
    while(i<n)
    {
        j=i+1;
        while((j<=n) && (a[i]==a[j])) j++;
        s++;
        i=j;
    }
    printf("\n\n%d\n", s);
    return 0;
}

```

Presupunând că tabloul  $a$  este ordonat crescător, precizați ce afișează programul dat:

- a) numărul valorilor distincte din tabloul  $a$ ;
- b) lungimea maximă a unei secvențe din tabloul  $a$  formată din valori egale;
- c) numărul secvențelor strict crescătoare din tabloul  $a$ ;
- d) lungimea maximă a unui subșir din tabloul  $a$  format din valori egale.

10. Se consideră următoarea funcție recursivă, scrisă în limbajul C:

```

int F(int n)
{
    if (n==0) return 0;
}

```

```

else
    if ((n%10)>F(n/10)) return n%10;
    else return F(n/10);
}

```

Ce valoare va returna funcția după apelul  $F(38423)$ ?

- a) 3
- b) 2
- c) 8
- d) 4

11. Se consideră următorul program în limbajul C:

```

#include<stdio.h>
int F(int v[],int n)
{
    if(n==0) return v[0];
    else
        if(v[n]<0) return F(v,n-1);
        else return v[n]+F(v,n-1);
}
int main()
{
    int i,v[10];
    for(i=0;i<10;i++)
        if(i%2==0) v[i]=i;
        else v[i]=-i;
    printf("%d",F(v,9));
    return 0;
}

```

Ce afișează programul de mai sus?

- a) -20
- b) 0
- c) 20
- d) -5

12. Se consideră următoarea funcție recursivă, scrisă în limbajul C:

```

int f(int n)
{
    if (n==0) return 1;
    else return(((n%10)%2 == 0) && (f(n/10)!=0));
}

```

Ce valoare va returna funcția după apelul  $f(6904)$ ?

- a) 4
- b) 6



- c) 9
- d) 0

13. Se consideră următoarea funcție recursivă, scrisă în limbajul C:

```
int f(int x)
{
    if(x==0) return 0;
    else return (f(x-1)+3*x-1);
}
```

Pentru ce valoare a parametrului  $x$  funcția  $f$  va întoarce valoarea 57?

- a) 5
- b) 6
- c) 8
- d) 10

14. Se consideră următoarea funcție recursivă, scrisă în limbajul C:

```
int p(int n, int x)
{
    if(x==n) return 1;
    else
        if(n%x==0) return 0;
        else return p(n, x+1);
}
```

În urma apelului  $p(n, 2)$  funcția va întoarce valoarea 1 dacă și numai dacă:

- a) numărul natural  $n$  este par;
- b) numărul natural  $n$  este prim;
- c) numărul natural  $n$  nu este prim;
- d) numărul natural  $n$  este impar.

15. Indicați care dintre următoarele metode de sortare se bazează pe tehnica de programare Divide et Impera:

- a) sortarea rapidă;
- b) sortarea prin interschimbare;
- c) sortarea prin interclasare;
- d) sortarea prin numărare.

16. Stabiliți care dintre următoarele metode de sortare nu se bazează pe tehnica de programare Divide et Impera:

- a) sortarea rapidă;
- b) sortarea prin interschimbare;

- c) sortarea prin interclasare;
- d) sortarea prin numărare.

17. Indicați care dintre următoarele metode de sortare au complexitatea  $\mathcal{O}(n \log_2 n)$ :

- a) sortarea rapidă;
- b) sortarea cu ansamble;
- c) sortarea prin interclasare;
- d) sortarea prin numărare.

18. Considerăm următoarele două funcții scrise în limbajul C:

```
int a[100];
int max(int x, int y)
{
    if(x>y) return x;
    else return y;
}
int F(int p, int u)
{
    if(p==u) return a[p];
    else
    {
        int m=(p+u)/2;
        return max(F(p,m), F(m+1,u));
    }
}
```

Știind că tabloul  $a$  este format din  $n$  numere naturale nenule, iar apelul subprogramului va fi  $F(0, n-1)$ , precizați tehnica de programare utilizată în cadrul funcției  $F$ :

- a) Greedy;
- b) backtracking;
- c) programarea dinamică;
- d) Divide et Impera.

19. Fie  $v$  un tablou unidimensional format din 1000000 de numere reale ordonate descrescător și  $x$  un număr real. Pentru a verifica dacă valoarea  $x$  se găsește sau nu în tabloul  $v$ , algoritmul de căutare binară va efectua:

- a) exact 1000000 de comparații;
- b) cel puțin 24 de comparații;
- c) cel mult 24 de comparații;
- d) nu se poate folosi algoritmul de căutare binară în acest caz.

20. Considerăm următoarea funcție scrisă în limbajul C:

```
int S(int a[], int p, int u)
{
    if(p>u) return 0;
```

```

else
{
    int m=(p+u)/2;
    return a[m] + S(a,p,m-1) + S(a,m+1,u);
}
}

```

Știind că tabloul  $a$  este format din  $n$  numere întregi, iar apelul subprogramului va fi  $S(a, 0, n - 1)$ , precizați ce va calcula funcția  $S$ :

- a) valoarea elementului din mijlocul tabloului  $a$ ;
- b) dublul sumei valorilor din tabloul  $a$ ;
- c) numărul valorilor pozitive din tabloul  $a$ ;
- d) **suma valorilor din tabloul  $a$ .**

21. Dacă ultima soluție afișată de algoritmul backtracking pentru generarea tuturor permutărilor mulțimii  $\{1, 2, \dots, 7\}$  este 7,6,3,5,4,2,1, atunci următoarea soluție care va fi afișată este:

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

22. Dacă ultima soluție afișată de algoritmul backtracking pentru generarea tuturor permutărilor mulțimii  $\{1, 2, \dots, 7\}$  este 6,5,7,4,3,2,1, atunci următoarea soluție care va fi afișată este:

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

23. Dacă ultima soluție afișată de algoritmul backtracking pentru generarea tuturor permutărilor mulțimii  $\{1, 2, \dots, 7\}$  este 6,7,4,5,3,2,1, atunci următoarea soluție care va fi afișată este:

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

24. Folosind tehnica de programare backtracking pentru a genera toate permutările mulțimii  $\{1, 2, \dots, n\}$ , o soluție se memorează sub forma unui tablou unidimensional  $x_1, x_2, \dots, x_n$ . Dacă au fost deja generate valori pentru componentele  $x_1, x_2, \dots, x_{k-1}$ , iar pentru componenta  $x_k$  ( $1 < k < n$ ) au fost deja testate toate valorile posibile și nu a fost găsită niciuna convenabilă, atunci:

- a) **se încearcă alegerea unei noi valori pentru  $x_{k-1}$ ;**
- b) se încearcă alegerea unei noi valori pentru  $x_1$ , oricare ar fi valoarea lui  $k$ ;

- c) se încheie algoritmul;
  - d) se încearcă alegerea unei valori pentru componenta  $x_{k+1}$ .
25. Considerăm ecuația  $a_1x_1 + a_2x_2 + \dots + a_nx_n = y$ , unde  $a_1, a_2, \dots, a_n, y$  sunt numere naturale nenule. Pentru a determina toate soluțiile ecuației de forma  $(x_1, x_2, \dots, x_n)$ , cu  $x_1, x_2, \dots, x_n$  numere naturale, se poate folosi direct algoritmul backtracking pentru:
- a) generarea permutărilor;
  - b) descompunerea unui număr natural ca sumă de numere naturale nenule;
  - c) plata unei sume folosind  $n$  tipuri de monede;
  - d) generarea combinărilor.
26. Un algoritm optim care să afișeze toate subșirurile crescătoare de lungime maximă ale unui șir format din  $n$  numere folosește:
- a) doar metoda programării dinamice;
  - b) doar metoda backtracking (se generează toate subșirurile șirului respectiv, iar pentru fiecare subșir se verifică dacă este crescător și, respectiv, maximal);
  - c) mai întâi metoda programării dinamice pentru a determina lungimea maximă  $lmax$  a unui subșir crescător al șirului dat și apoi metoda backtracking pentru a genera toate subșirurile crescătoare de lungime  $lmax$  ale șirului considerat;
  - d) doar metoda Greedy.
27. Considerăm că în Facultatea de Informatică sunt înscriși  $n$  studenți în anul III. Pentru a afișa toate grupele ce pot fi formate din câte  $p$  studenți ( $p \leq n$ ) putem folosi algoritmul backtracking pentru:
- a) generarea aranjamentelor formate din  $p$  elemente ale unei mulțimi cu  $n$  elemente;
  - b) generarea permutărilor unei mulțimi cu  $p$  elemente;
  - c) generarea combinărilor formate din  $p$  elemente ale unei mulțimi cu  $n$  elemente;
  - d) generarea aranjamentelor formate din  $n$  elemente ale unei mulțimi cu  $p$  elemente.
28. Utilizând metoda backtracking, se generează toate descompunerile distincte ale numărului natural  $n = 10$  ca sumă a cel puțin două numere naturale nenule. Care este ultima descompunere generată?
- a)  $4 + 3 + 2 + 1$
  - b)  $9 + 1$
  - c)  $4 + 6$
  - d)  $5 + 5$
29. Utilizând metoda backtracking, se generează toate descompunerile distincte ale numărului natural  $n = 16$  ca sumă a cel puțin două numere naturale nenule. Care este ultima descompunere generată?
- a)  $4 + 4 + 4 + 4$

- b)  $15 + 1$   
 c)  $7 + 9$   
 d)  $8 + 8$
30. Fie  $S$  o sumă de bani și  $v_1, v_2, \dots, v_n$  valorile a  $n$  tipuri de monede. Presupunând că din fiecare tip avem la dispoziție un număr nelimitat de monede, pentru afișarea tuturor modalităților în care poate fi plătită suma  $S$  folosind monede disponibile trebuie să utilizăm un algoritm bazat pe metoda:
- a) Greedy;  
 b) **backtracking**;  
 c) programării dinamice;  
 d) Divide et Impera.
31. Considerăm un rucsac cu ajutorul căruia putem transporta 66 kg și 7 obiecte având greutatea 23, 10, 10, 25, 38, 7 și 5 kg, iar câștigurile obținute prin transportul integral al fiecărui obiect la destinație sunt 69, 10, 30, 100, 19, 14 și 50 RON. Știind că din orice obiect putem încărca și numai o parte a sa, câștigul maxim pe care îl putem obține este:
- a) 250.5 RON  
 b) 217 RON  
 c) 265 RON  
 d) **255 RON**
32. Considerăm un rucsac cu ajutorul căruia putem transporta 67 kg și 7 obiecte având greutatea 10, 5, 20, 10, 20, 25 și 21 kg, iar câștigurile obținute prin transportul integral al fiecărui obiect la destinație sunt 30, 40, 40, 10, 4, 50 și 30 RON. Știind că din oricare obiect putem încărca și numai o parte a sa, câștigul maxim pe care îl putem obține este:
- a) 114 RON  
 b) **170 RON**  
 c) 280 RON  
 d) 163.7 RON
33. Considerăm un rucsac cu ajutorul căruia putem transporta 53 kg și 7 obiecte având greutatea 10, 5, 18, 10, 8, 20 și 40 kg, iar câștigurile obținute prin transportul integral al fiecărui obiect la destinație sunt 30, 40, 36, 10, 16, 10 și 30 RON. Știind că din oricare obiect putem încărca și numai o parte a sa, câștigul maxim pe care îl putem obține este:
- a) 133 RON  
 b) 121 RON  
 c) **133.5 RON**  
 d) 136.5 RON
34. Stabiliți care dintre următoarele propoziții referitoare la tehnica de programare Greedy sunt adevărate:

- a) conduce întotdeauna la o soluție optimă;
  - b) construiește o soluție element cu element și în cazul în care valoarea elementului curent nu verifică anumite condiții se renunță la acesta și se revine la elementul anterior;
  - c) găsește întotdeauna o singură soluție a unei probleme;
  - d) conduce la o soluție optimă doar în cazul în care s-a demonstrat matematic corectitudinea criteriului de selecție pe baza căruia un element din mulțimea inițială este adăugat în mulțimea ce reprezintă soluția problemei.
35. La un ghișeu stau la coadă  $n$  persoane, numerotate cu  $1, 2, \dots, n$ . Cunoscând timpii de servire  $t_1, t_2, \dots, t_n$  ai celor  $n$  persoane și știind că pentru a servi o persoană  $k$  trebuie servite persoanele  $1, 2, \dots, k - 1$  aflate înaintea sa, trebuie să determinăm un mod de rearanjare al persoanelor la coadă, astfel încât timpul de așteptare al fiecărei persoane să fie minim. Stabiliți care dintre următoarele variante de rezolvare a acestei probleme este corectă și are o complexitate minimă:
- a) se generează toate modurile în care pot fi rearanjate cele  $n$  persoane la coadă și pentru fiecare mod de rearanjare se calculează într-un tablou timpii de servire, iar soluția este dată de tabloul minim în sens lexicografic;
  - b) se rearanjează persoanele în ordinea descrescătoare a timpilor de servire;
  - c) se generează toate modurile în care pot fi rearanjate cele  $n$  persoane la coadă și pentru fiecare mod de rearanjare se calculează timpul total  $T$  de servire al celor  $n$  persoane, iar soluția este tabloul pentru care valoarea lui  $T$  este minimă;
  - d) se rearanjează persoanele în ordinea crescătoare a timpilor de servire.
36. La un ghișeu stau la coadă 7 persoane  $p_1, p_2, \dots, p_7$ . Cunoscând timpii lor de servire  $t_1 = 10$ ,  $t_2 = 6$ ,  $t_3 = 5$ ,  $t_4 = 6$ ,  $t_5 = 2$ ,  $t_6 = 8$ ,  $t_7 = 4$  și știind că pentru a servi o persoană trebuie servite, mai întâi, toate persoanele aflate înaintea sa, precizați care dintre următoarele modalități de rearanjare a persoanelor la coadă minimizează timpul mediu de așteptare:
- a)  $p_5, p_7, p_3, p_2, p_4, p_6, p_1$
  - b)  $p_1, p_6, p_4, p_2, p_3, p_5, p_7$
  - c)  $p_5, p_7, p_3, p_4, p_2, p_6, p_1$
  - d)  $p_3, p_7, p_2, p_1, p_6, p_4, p_5$
37. La un ghișeu stau la coadă 7 persoane  $p_1, p_2, \dots, p_7$ . Cunoscând timpii lor de servire  $t_1 = 12$ ,  $t_2 = 6$ ,  $t_3 = 15$ ,  $t_4 = 16$ ,  $t_5 = 12$ ,  $t_6 = 8$ ,  $t_7 = 2$  și știind că pentru a servi o persoană trebuie servite, mai întâi, toate persoanele aflate înaintea sa, precizați care dintre următoarele modalități de rearanjare a persoanelor la coadă minimizează timpul mediu de așteptare:
- a)  $p_7, p_2, p_6, p_5, p_1, p_3, p_4$
  - b)  $p_3, p_7, p_2, p_1, p_6, p_4, p_5$
  - c)  $p_4, p_3, p_1, p_5, p_6, p_2, p_7$
  - d)  $p_7, p_2, p_6, p_1, p_5, p_3, p_4$

38. În Aula Magna a Universității Titu Maiorescu din București se va organiza un festival de teatru care va dura o singură zi. Fiecare regizor a transmis organizatorului festivalului intervalul de timp în care se poate desfășura spectacolul său. Organizatorul festivalului dorește să programeze un număr maxim de spectacole. Știind că spectacolele nu se pot suprapune și că între oricare două spectacole consecutive nu există nicio pauză, stabiliți care dintre strategiile de planificare de tip Greedy de mai jos pot fi folosite de către organizatorul festivalului pentru a planifica un număr maxim de spectacole în Aula Magna în ziua respectivă:
- se sortează spectacolele în ordinea crescătoare a orelor la care se termină, se programează primul spectacol și apoi se consideră, pe rând, restul spectacolelor, un spectacol fiind programat doar dacă începe după ce se termină spectacolul programat anterior;
  - se sortează spectacolele în ordinea crescătoare a duratei lor, se programează primul spectacol și apoi se consideră, pe rând, restul spectacolelor, un spectacol fiind programat doar dacă începe după ce se termină spectacolul programat anterior;
  - se sortează spectacolele în ordinea crescătoare a orelor la care încep, se programează primul spectacol și apoi se consideră, pe rând, restul spectacolelor, un spectacol fiind programat doar dacă începe după ce se termină spectacolul programat anterior;
  - se sortează spectacolele în ordinea descrescătoare a orelor la care se termină, se programează primul spectacol și apoi se consideră, pe rând, restul spectacolelor, un spectacol fiind programat doar dacă începe după ce se termină spectacolul programat anterior.
39. În Aula Magna a Universității Titu Maiorescu din București se va organiza un festival de teatru care va dura o singură zi. Fiecare regizor a transmis organizatorului festivalului intervalul de timp  $[hh_1:mm_1, hh_2:mm_2]$  în care se poate desfășura spectacolul său. Știind că spectacolele nu se pot suprapune și între oricare două spectacole consecutive nu există nicio pauză, organizatorul festivalului s-a gândit să folosească o strategie de planificare de tip Greedy pentru a planifica un număr maxim de spectacole în cadrul festivalului. Considerând că 7 regizori au trimis intervalele de desfășurare ale spectacolelor lor  $s_1 = [08:00, 10:30]$ ,  $s_2 = [08:30, 09:00]$ ,  $s_3 = [16:30, 18:00]$ ,  $s_4 = [10:30, 10:45]$ ,  $s_5 = [11:00, 17:00]$ ,  $s_6 = [11:30, 13:00]$ ,  $s_7 = [15:15, 16:45]$ , precizați care dintre variantele de mai jos reprezintă o planificare corectă, cu un număr maxim de spectacole:
- S1, S2, S4, S5, S6
  - S2, S4, S6, S7
  - S2, S4, S6, S3
  - S1, S5
40. În Aula Magna a Universității Titu Maiorescu din București se va organiza un festival de teatru care va dura o singură zi. Fiecare regizor a transmis organizatorului festivalului intervalul de timp  $[hh_1:mm_1, hh_2:mm_2]$  în care se poate desfășura spectacolul său. Știind că spectacolele nu se pot suprapune și între oricare două spectacole consecutive nu există nicio pauză, organizatorul festivalului s-a gândit să folosească o strategie de planificare de tip Greedy pentru a planifica un număr maxim de spectacole în cadrul festivalului. Considerând că 7 regizori au trimis intervalele de desfășurare ale spectacolelor lor

$s_1 = [08:00, 10:30]$ ,  $s_2 = [08:30, 11:00]$ ,  $s_3 = [16:30, 18:00]$ ,  $s_4 = [10:30, 10:45]$ ,  $s_5 = [11:00, 12:00]$ ,  $s_6 = [12:30, 16:00]$ ,  $s_7 = [17:15, 18:45]$ , precizați care dintre variantele de mai jos reprezintă o planificare corectă, cu un număr maxim de spectacole:

- a) S1, S5, S6, S3
- b) S2, S4, S5, S6, S7
- c) S1, S4, S5, S6, S3
- d) S1, S4, S5, S6, S7

41. Se consideră un triunghi de numere întregi format din  $n$  linii, astfel: prima linie conține un număr, a doua linie conține două numere, . . . , ultima linie conține  $n$  numere. În acest triunghi se pot forma sume de numere întregi în felul următor:

- se selectează numărul aflat pe prima linie;
- la fiecare pas se selectează fie numărul aflat pe următoarea linie sub ultimul număr selectat, fie numărul aflat pe următoarea linie și o coloană la dreapta față de ultimul număr selectat, până când se ajunge pe ultima linie a triunghiului de numere.

Un algoritm cu complexitate minimă care determină cea mai mare sumă ce se poate obține respectând regulile de mai sus folosește metoda:

- a) Greedy;
- b) backtracking;
- c) programării dinamice;
- d) Divide et Impera.

42. Se consideră următorul triunghi de numere întregi format din  $n = 4$  linii:

```

10
12  -20
30   50  180
21   14   20  16

```

În acest triunghi se pot forma sume de numere întregi în felul următor:

- se selectează numărul aflat pe prima linie;
- la fiecare pas se selectează fie numărul aflat pe următoarea linie sub ultimul număr selectat, fie numărul aflat pe următoarea linie și o coloană la dreapta față de ultimul număr selectat, până când se ajunge pe ultima linie a triunghiului de numere.

Care este suma maximă ce poate fi obținută în triunghiul dat, respectând condițiile precizate mai sus?

- a) 190
- b) 73
- c) 92
- d) 302

43. Se consideră următorul triunghi de numere întregi format din  $n = 4$  linii:



```

100
-12  -20
130  200  118
210  114  320  160

```

În acest triunghi se pot forma sume de numere întregi în felul următor:

- se selectează numărul aflat pe prima linie;
- la fiecare pas se selectează fie numărul aflat pe următoarea linie sub ultimul număr selectat, fie numărul aflat pe următoarea linie și o coloană la dreapta față de ultimul număr selectat, până când se ajunge pe ultima linie a triunghiului de numere.

Care este suma maximă ce poate fi obținută în triunghiul dat, respectând condițiile precizate mai sus?

- a) 518
- b) 402
- c) 428
- d) 608

44. Fie  $S$  o sumă de bani și  $v_1, v_2, \dots, v_n$  valorile a  $n$  tipuri de monede (se presupune că din fiecare tip de monedă avem la dispoziție un număr nelimitat de monede). Un algoritm optim care să determine numărul minim de monede cu care poate fi plătită suma  $S$ , folosind monede de tipurile date, folosește metoda:

- a) Greedy;
- b) backtracking;
- c) programării dinamice;
- d) Divide et Impera.

45. Precizați câte subșiruri strict crescătoare de lungime maximă conține tabloul  $v = (12, 18, 17, 11, 10, 16, 17, 26)$ :

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

46. Precizați câte subșiruri strict crescătoare de lungime maximă conține tabloul  $v = (21, 18, 13, 19, 21, 16, 16, 17, 26)$ :

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

47. Indicați lungimea maximă a unui subșir strict crescător din tabloul  $v = (12, 18, 17, 11, 10, 26, 17, 26, 19, 28)$ :
- a) 2
  - b) 4
  - c) 3
  - d) 5
48. Indicați lungimea maximă a unui subșir strict crescător din tabloul  $v = (18, 14, 17, 4, 8, 15, 12, 19, 16, 22)$ :
- a) 2
  - b) 4
  - c) 3
  - d) 5
49. Având la dispoziție un număr nelimitat de monede cu valorile 5 RON, 4 RON, 3 RON și 1 RON, precizați numărul minim de monede cu care poate fi plătită suma de 17 RON:
- a) 4
  - b) 5
  - c) 3
  - d) 6
50. Având la dispoziție un număr nelimitat de monede cu valorile 7 RON, 6 RON, 3 RON și 2 RON, precizați numărul minim de monede cu care poate fi plătită suma de 19 RON:
- a) 6
  - b) 3
  - c) 4
  - d) 5

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**MODULUL 2**

***Baze de date***

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Baze de date**

**&**

**Disciplina Sisteme de gestiune a bazelor de date**

## Subiecte BD

1. O bază de date reprezintă:

- A. Un ansamblu de posturi de lucru, fișiere și unități de prelucrare
- B. Un ansamblu de articole secvențiale accesate direct
- C. Un ansamblu de înregistrări accesate în paralel
- D. Un ansamblu integrat de înregistrări sau de fișiere interconectate în mod logic**
- E. Suportul pentru înregistrarea datelor

2. Depozitul de date reprezintă:

- A. O colecție de informații privind un sector dintr-o întreprindere sau o firmă
- B. O bază de date împreună cu SGBD-ul asociat
- C. O colecție de date organizate secvențial
- D. O bază de date special concepută prin agregarea tuturor datelor dintr-o organizație sau întreprindere, în vederea sprijinirii procesului de luare a deciziilor**
- E. O bibliotecă de informații privind un anumit sector economic

3. Nivelul fizic de abstractizare a datelor permite:

- A. Utilizarea concretă a posibilităților SGBD-ului
- B. Proiectarea dicționarului de date
- C. Asigurarea securității datelor
- D. Automatizarea operațiilor de întreținere și dezvoltare a programelor
- E. Descrierea structurii sub care datele se regăsesc pe suportul de memorare**

4. Când se recomandă să se introducă datele într-o baza de date?

- A. După crearea fiecărei tabele în parte
- B. După crearea tuturor tabelor și definirea restricțiilor**
- C. După crearea tuturor tabelor
- D. Oricând
- E. Datele sunt introduse de către administrator. Utilizatorii doar interogază baza de date

5. Care dintre următoarele enunțuri caracterizează modelul relațional?

- A. Este singurul model în care structura datelor este definită prin normalizare
- B. Asigură independența dintre date și prelucrări
- C. Este fundamentat pe teoria matematică a relațiilor**
- D. Respectă restricțiile referențiale
- E. Trebuie să existe cel puțin o cheie secundară

6. Intersecția a două relații:

- A. Are ca rezultat o relație cu un număr dublu de linii
- B. Are ca rezultat o relație cu un număr de două ori mai mic de linii
- C. Este o operație unară
- D. Cere ca acestea să aibă aceeași schemă**
- E. Va avea ca rezultat o relație cu gradul egal cu suma gradelor relațiilor inițiale

7. Se dă tabelul:

ANGAJATI (CodAngaj, Nume, DataAngajare, Compartiment, Profesia, Salariu)  
Care dintre următoarele instrucțiuni (Transact) SQL permite afișarea numărului de salariați pentru fiecare compartiment care începe cu litera "J"?

- A. **SELECT COUNT(CodAngajat) AS Numar, Compartiment FROM ANGAJATI WHERE Compartiment LIKE 'J%' GROUP BY Compartiment**
- B. SELECT CodAngajat, Nume, Compartiment FROM ANGAJATI WHERE Nume LIKE 'J%'
- C. SELECT COUNT(CodAngajat) AS Numar, Nume, Compartiment FROM ANGAJATI WHERE Compartiment LIKE 'J%' GROUP BY Compartiment
- D. SELECT CodAngajat, Nume, Compartiment FROM ANGAJATI WHERE Compartiment LIKE 'J%'
- E. Niciuna dintre cele de mai sus

8. Ce se va obține prin următoarea interogare (Transact) SQL:

SELECT Nume FROM STUDENTI INNER JOIN LOCALIT  
ON STUDENTI.CodLoc=LOCALIT.CodLoc  
WHERE CodLoc IN

(SELECT CodLoc FROM STUDENTI WHERE Nume = 'Ionescu M Petre') ?

- A. Numele studenților asemănătoare cu Ionescu M Petre
- B. Codul localității din care este studentul Ionescu M Petre
- C. Numărul studenților care sunt din aceeași localitate cu Ionescu M Petre
- D. Fraza SELECT este scrisă greșit
- E. **Numele studenților care sunt din aceeași localitate cu Ionescu M Petre**

9. Se dă următoarea instrucțiune (Transact) SQL:

SELECT NrFactura  
FROM CLIENTI INNER JOIN FACTURIEMISE ON  
CLIENTI.CodClient=FACTURIEMISE.Codclient  
WHERE ValTot IN (100, 200)

În urma execuției acestei interogări se obțin:

- A. Numai facturile emise cu valoare între 100 și 200, exclusiv capetele
- B. Numai facturile emise cu valoare între 100 și 200, inclusiv aceste valori
- C. **Numai facturile emise cu valoarea de 100 sau 200**
- D. Numai facturile emise cu valoare fie mai mică de 100, fie mai mare de 200
- E. Nimic, interogarea este greșită

10. Se dă tabelul:

CHITANTE (NrChitanta, Suma, Data, NrGhiseu )

Care dintre următoarele instrucțiuni (Transact) SQL are ca efect afișarea zilelor în care s-au emis cel puțin 3 chitanțe?

A. SELECT NrChitanta, Suma, Data FROM CHITANTE WHERE COUNT(NrChitanta) >= 3

B. SELECT NrChitanta, Suma, Data FROM CHITANTE HAVING COUNT(NrChitanta) >= 3

C. SELECT COUNT(Data), COUNT(NrChitanta) FROM CHITANTE WHERE Data IS NULL

D. SELECT Data FROM CHITANTE WHERE COUNT(NrChitanta) >= 3

**E. SELECT CHITANTE.Data FROM CHITANTE GROUP BY CHITANTE.Data HAVING COUNT(NrChitanta) >= 3**

11 Se dă tabelul:

ANGAJATI (CodAngaj, Nume, DataAngajare, Compartiment, Profesia, Salariu)

Care dintre următoarele instrucțiuni (Transact) SQL permite majorarea salariului cu 3% pentru salariații angajați în anul 2010?

A. UPDATE Angajati SET Salariu = Salariu + 3% WHERE DataAngajare IN 2010

B. UPDATE Salariu FROM Angajati SET Salariu = Salariu + 3% \* Salariu WHERE DataAngajare IN ('1/1/2010', '31/12/2010')

**C. UPDATE Angajati SET Salariu = Salariu \* 1.03 WHERE YEAR(DataAngajare) = 2010**

D. UPDATE Salariu FROM Angajati SET Salariu = Salariu \* 1.03 WHERE DataAngajare BETWEEN '1/1/2010' AND '31/12/2010'

E. Niciuna

12. Se dă entitatea ANGAJATI, cu următoarele atribute:

ANGAJATI (CodAngaj, Nume, DataAngajare, Compartiment, Profesia, Salariu)

Care dintre următoarele instrucțiuni (Transact) SQL permite ștergerea salariaților angajați înainte de începutul anului 2010, cu profesia "informatician"?

A. DELETE FROM Angajati WHERE DataAngajare < '1/12/2009' AND Profesia LIKE '%informatician%'

B. DELETE \* FROM Angajati WHERE Profesia LIKE 'informatician' HAVING YEAR(DataAngajare) < 2010

C. UPDATE Angajati SET CodAngaj = Null WHERE DataAngajare < '1/1/2010' AND Profesia LIKE '%informatician%'

D. Niciuna

**E. DELETE FROM Angajati WHERE DataAngajare < '1/1/2010' AND Profesia LIKE '%informatician%'**

13. Care dintre următoarele comenzi SQL sunt destinate manipulării datelor?

- A. INSERT
- B. CREATE TABLE
- C. UPDATE
- D. ROLLBACK
- E. DELETE
- F. COMMIT

14. Pentru o baza de date relațională algebra relațională este:

- A. Completa (permite să extrag orice informații din bază)
- B. Procedurală (Arată și modul cum se obțin informațiile din baza de date)
- C. Complexă (permite să extrag informații obiectuale)
- D. Limbaj formal
- E. Limbaj de definire a rapoartelor
- F. Limbaj de interogare

15. Când creai o bază de date, care sunt fișierele care trebuie create obligatoriu?

- A. Un fișier de date primar
- B. Un fișier de date primar și un fișier de log
- C. Un fișier de date primar, un fișier de date secundar și un fișier de log
- D. Un fișier de log primar

16. Folosind tabela cadru\_did(cod, nume, graddid, data\_ang, sal\_neg, sal\_merit), creată și populată cu date, se consideră următoarele exemple:

1. SELECT graddid, max(sal\_neg)  
FROM cadru\_did  
WHERE max(sal\_neg) > 1600  
GROUP BY graddid
2. SELECT graddid, max(sal\_neg)  
FROM cadru\_did  
GROUP BY graddid  
HAVING max(sal\_neg) > 1600

Să se aleagă răspunsul corect:

- A. Exemplul (1) este corect
- B. Exemplul (2) este corect
- C. Exemplul (2) nu este corect
- D. Exemplele (1) și (2) sunt corecte
- E. Exemplul (1) nu este corect
- F. Nici un exemplu nu este corect



17. Cu ajutorul comenzii SELECT :

- A. Se poate realiza joncțiunea externă(outer-join) între tabele
- B. Se poate realiza joncțiunea de egalitate(equi-join) între tabele
- C. Nu se pot realiza joncțiuni între tabele
- D. Se poate realiza joncțiunea de neegalitate(non-equi-join) între tabele
- E. Se poate modifica structura unei tabele
- F. Se poate lega o tabelă cu ea însăși

18. Dându-se tabelele *Produse*(**codp**, denp, pret ) și *Comenzi*(**nr\_cda**, **codp**, cant, data), următoarele interogări:

1. SELECT c.nr\_cda, c.codp, c.cant FROM comenzi c  
WHERE 'penar'=(SELECT den\_prod FROM produse p WHERE p.codp=c.codp)
2. SELECT c.nr\_cda, c.codp, c.cant  
FROM comenzi c JOIN produse p  
ON p.codp=c.codp WHERE p.den\_prod='penar'

- A. Sunt identice ca efect
- B. Sunt diferite ca efect
- C. Ar fi identice ca efect dacă în cererea 2 s-ar folosi produsul cartezian
- D. Ar fi identice ca efect dacă în cererea 1 s-ar folosi o subinterogare necorelata

19. Folosind relația *cadru\_did*(cod, nume, graddid, data\_ang, sal\_neg, sal\_merit), să se specifice care dintre următoarele exemple sunt corecte:

- A. SELECT avg(sal\_neg) FROM cadru\_did
- B. SELECT graddid, sum(sal\_neg) FROM cadru\_did
- C. SELECT graddid, avg(sal\_neg) FROM cadru\_did GROUP BY graddid
- D. SELECT graddid, sum(sal\_neg), count(cod) FROM cadru\_did GROUP BY data\_ang
- E. SELECT graddid, avg(sal\_neg) FROM cadru\_did WHERE avg(sal\_neg) > 1800

20. Interogarea:

```
SELECT min(nume), sum(nume), count(*)  
FROM tabela  
WHERE salariu = max(salariu)
```

- A. Este greșită deoarece funcția min() nu se folosește cu șiruri de caractere
- B. Este greșită deoarece funcția sum() nu se folosește cu șiruri de caractere
- C. Este corectă
- D. Este greșită deoarece funcția max() nu se folosește în clauza WHERE
- E. Este greșită deoarece nu se folosește \* în funcția count()

21. Se poate folosi același nume pentru mai multe tabele dintr-o bază de date Microsoft SQL Server?

- A. Da, dar să nu aparțină aceleleași baze de date
- B. Da, dacă nu fac parte din aceeași schemă
- C. Nu
- D. Da, dacă aparțin aceleleași scheme
- E. Da, dar să nu aparțină aceleleași instanțe

22. Interogările din SQL sunt la fel de expresive cu:

- A. Algebra relationala
- B. Calculul relational al domeniilor
- C. Calculul procedural
- D. Calculul relational al tuplurilor
- E. Calculul bazat pe tablouri

23. Avem apriori o multitudine de restricții, dată de o mulțime de dependențe funcționale  
F. Aceasta este folosită la:

- A. Normalizarea bazei de date
- B. Crearea bazei de date
- C. Crearea procedurilor de interogare
- D. La validarea stării bazei de date
- E. Construcția algebrei relaționale

24. Fie tabelul METEO(localit varchar(30), tempmin numeric(3), tempmax numeric(4)).  
Să se stabilească în care dintre exemplele următoare se folosesc incorect operatorii  
aritmetici:

- A. SELECT localit, -tempmin minim, +tempmax maxim FROM meteo
- B. SELECT localit, (tempmax-tempmin) diferenta FROM meteo
- C. SELECT -localit, -tempmin FROM meteo
- D. SELECT localit, (tempmax-tempmin)-1 FROM meteo
- E. SELECT localit\*2, tempmax\*2 FROM meteo

25. Operatorul NOT se poate folosi cu:

- A. Operatorul LIKE
- B. Operatorul IS NULL
- C. Operatorul >
- D. Operatorul IN (lista)
- E. Operatorul <>

26. Fragmentarea verticală se obține dintr-o relație prin:

- A. Selecții și replicări
- B. Selecții și proiecții
- C. Proiecție
- D. Unire (Join)
- E. Introducerea de chei externe

27. Se dau tabelele:

PERSONAL(Marca, Numepren, Compart, Datasv, Salorar, Salorarco, Compart)

PONTAJE(Marca, Data, Orelucrate, Oreco, Orenoapte, Oreabsnem)

Ce afișează următoarea interogare?

SELECT ZILE1.Data AS Ziua

FROM (SELECT Data FROM PONTAJE po INNER JOIN PERSONAL pe ON

po.Marca=pe.Marca WHERE Numepren = 'Popescu') ZILE1

INNER JOIN (SELECT Data FROM PONTAJE po INNER JOIN PERSONAL pe ON

po.Marca =pe.Marca WHERE Numepren='Georgescu') ZILE2

ON ZILE1.Data=ZILE2.Data

A. Angajatul care a lucrat mai multe zile

B. Numărul total de zile în care au lucrat cei doi angajați

C. Numărul de zile lucrate de fiecare angajat

**D. Zilele în care au lucrat simultan cei doi angajați**

E. Numărul de zile în care nu au lucrat cei doi angajați

28. Se dau tabelele:

ANGAJATI(ID\_Angajat, Nume, ID\_Compart, Salariu)

COMPARTIMENT (ID\_Compart, Nume\_Compart)

și următoarea interogare:

SELECT a.ID\_Angajat, a.Nume, c.ID\_Compart, c. Nume\_Compart

FROM ANGAJATI a, COMPARTIMENT c

WHERE a. ID\_Compart = c. ID\_Compart

Ce operații implementează această interogare?

A. Diferența, proiecția și join

B. Selecția, intersecția și join

C. Intersecția, proiecția și join

**D. Selecția, proiecția și join**

E. Diferența, intersecția și join

29. Firma la care lucrați folosește un tabel *Proiecte*(**Cod\_proiect**, tester,...), unde valoarea coloanei *tester* este un nr. întreg care corespunde codului angajatului din tabela *Angajati* (**Id\_ang**, nume,...) care a răspuns de testarea proiectului(responsabil unic).

Coloana *tester* are valoarea NULL dacă n-a fost desemnat nici un responsabil cu testarea proiectului respectiv. Trebuie să afișați numele angajaților din tabela *Angajati* care nu au condus nici un test pe un proiect. Ce instrucțiune SELECT ar trebui folosită?

A. SELECT nume FROM Angajati WHERE Id\_ang < >

(SELECT tester FROM Proiecte WHERE tester = Id\_ang )

**B. SELECT nume FROM Angajati WHERE NOT EXISTS**

**(SELECT tester FROM Proiecte WHERE tester = Id\_ang )**

C. SELECT nume FROM Angajati WHERE Id\_ang NOT IN

(SELECT tester FROM Proiecte)

D. SELECT nume FROM Angajati LEFT JOIN Proiecte

ON Id\_ang = tester

30. O bază de date distribuită este:

- A. O colecție de date împărțită în mai multe fragmente
- B. O colecție de date împărțită în mai multe replici
- C. O colecție de date și de scheme partajate și interconectate logic, distribuite fizic pe calculatoarele unei rețele
- D. O colecție de fragmente mixte
- E. Niciuna

31. Baza de date curentă include o tabelă numită *Employees* cu următoarele coloane:

- *firstName*, care păstrează prenumele unui angajat;
- *lastName*, care păstrează numele unui angajat;
- *depID*, care păstrează ID-ul departamentului în care lucrează angajații.

Ați cerut să se afișeze o listă cu angajații sortată descrescător după ID departament, apoi după numele angajaților (prenume urmat de nume) în ordinea alfabetică.

Ce instrucțiune SELECT trebuie să folosiți?

- A. 

```
SELECT depID, (firstName + ' ' + lastName) AS 'Full Name'
FROM Employees
ORDER BY depID, lastName ASC
```
- B. 

```
SELECT (firstName + ' ' + lastName) AS 'Full Name'
FROM Employees
ORDER BY depID DESC, 'Full Name'
```
- C. 

```
SELECT lastName, (firstName + ' ' + lastName) AS 'Full Name'
FROM Employees
ORDER BY lastName, depID
```
- D. 

```
SELECT (firstName + ' ' + lastName) AS 'Full Name'
FROM Employees
ORDER BY depID DESC, lastName
```
- E. 

```
SELECT (firstName + ' ' + lastName) AS 'Full Name'
FROM Employees
ORDER BY depID DESC, lastName DESC
```

32. Datorită mutării activității comerciale din Germania în SUA, trebuie să schimbăm *collation data* pentru baza de date *Adventure Works*. După modificarea *collation* pentru această bază de date, ce ar mai trebui să faceți?

- A. Nimic.
- B. Ștergeți indecșii și constrângerile tabelor existente pentru care *collation* cere să fie modificat. Modificați *collation* pentru coloanele acelor tabele. Recreați indecșii și constrângerile.
- C. Ștergeți indecșii și constrângerile pentru toate tabele existente din baza de date. Modificați *collation* pentru coloanele acelor tabele. Recreați indecșii și constrângerile.
- D. Modificați database *collation* și selectați opțiunea auto-indexing.
- E. Recreați toți indecșii în această bază de date.

33. Firma la care lucrați folosește o tabelă numită *Products* pentru a stoca informații despre produsele pe care le vinde. Tabela include coloanele:

- *price*, care conține prețul produsului vândut;
- *volume*, care păstrează volumul vânzărilor lunare medii pentru produsul respectiv;
- *stock*, care stochează numărul de bucăți aflate în stoc pentru un produs.

La începutul fiecărei luni, firma dvs. reface stocul unui produs dacă una dintre condițiile următoare este adevărată:

- Produsul costă 100\$ sau mai mult și numărul de bucăți din stoc este mai mic decât volumul lunar mediu sau mai mic de 500.

- Produsul costă mai puțin de 100 \$ și sunt mai puțin de 1500 de bucăți în stoc.

Dvs. trebuie să obțineți informații despre toate produsele care trebuie să fie realimentate.

Ce instrucțiune ar trebui folosită?

A. `SELECT * FROM Products`  
`WHERE`  
`price >= 100.00`  
`AND stock < 500`  
`OR stock < volume`  
`OR (price < 100.00`  
`AND stock < 1500)`

B. *`SELECT * FROM Products`*  
*`WHERE`*  
*`price >= 100.00`*  
*`AND (stock < 500`*  
*`OR stock < volume)`*  
*`OR price < 100`*  
*`AND stock < 1500`*

C. `SELECT * FROM Products`  
`WHERE`  
`(price >= 100.00`  
`AND stock < 500`  
`OR stock < volume)`  
`OR (price < 100.00`  
`AND stock < 1500)`

D. `SELECT * FROM Products`  
`WHERE`  
`price >= 100.00`  
`AND stock < 500`  
`OR stock < volume`  
`OR price < 100.00`  
`AND stock < 1500`

E. `SELECT * FROM Products`  
`WHERE`  
`price >= 100.00`  
`AND stock < 500`  
`OR stock < volume`  
`AND price < 100.00`

34. Pentru a afișa numele angajaților care au un salariu mai mare decât al cel puțin unui manager existent, ar trebui să folosiți instrucțiunea:

SELECT name FROM Employees

WHERE salary > ANY (SELECT salary FROM Management).

Trebuie să creați o instrucțiune alternativă care să dea același rezultat și care să se execute cel puțin la fel de rapid.

Ce instrucțiune ar trebui să folosiți ?

A. SELECT name FROM Employees

WHERE NOT salary < ALL (SELECT salary FROM Management)

B. SELECT name FROM Employees

WHERE NOT EXISTS

(SELECT \* FROM Management WHERE salary >= Employees.salary)

C. SELECT name FROM Employees

WHERE EXISTS

(SELECT \* FROM Management WHERE salary < Employees.salary)

D. SELECT name FROM Employees

JOIN Management ON Employees.salary > Management.salary

E. Nici una.

35. Aveți o tabelă numită *Employees*, care include următoarele coloane:

- *empName*, care conține numele angajatului;

- *department*, care conține numele departamentului în care lucrează angajatul.

Pentru fiecare angajat care lucrează într-un departament cu mai mult de 10 angajați, trebuie să afișați numele angajatului, numele departamentului și nr. de angajați din acel departament. Trebuie să scrieți o interogare care să extragă această informație în cel mai scurt timp. Ce instrucțiune ar trebui folosită ?

A. SELECT empName, department,

(SELECT COUNT(\*) FROM Employees WHERE department = e.department)

FROM Employees AS e

B. SELECT department, COUNT(\*) AS employees

INTO DepCounts

FROM Employees

GROUP BY department

SELECT e.empName, e.department, d.employees

FROM Employees AS e

JOIN DepCounts AS d ON e.department = d.department

WHERE d.employees > 10

C. SELECT empName, department,

(SELECT COUNT(\*) FROM Employees WHERE department = e.department)

FROM Employees AS e WHERE 10 < (SELECT COUNT(\*) FROM Employees

WHERE department = e.department);

D. SELECT empName, department,

(SELECT COUNT(\*) FROM Employees WHERE (department = e.department) And count(\*) > 10) FROM Employees AS e

E. Nici una.

36. Ați scris o instrucțiune SELECT complexă care afișează date din mai multe tabele. Trebuie să creați o tabelă nouă și să o populați cu aceste date folosind cel mai mic număr de instrucțiuni posibil. Ce ar trebui să faceți ?

- A. Creați tabela și apoi folosiți instrucțiunea INSERT INTO ... SELECT FROM pentru a o popula.
- B. Folosiți o instrucțiune SELECT INTO pentru a crea și popula tabela.
- C. Folosiți o instrucțiune SELECT INTO pentru a crea și popula o tabelă temporară, apoi creați tabela permanentă din tabela temporară.
- D. Creați tabela, apoi folosiți o instrucțiune SELECT INTO pentru a o popula.
- E. Folosiți două proceduri stocate: una pentru crearea tabelului și alta pentru populare.

37. Firma la care lucrați vinde echipamente de la fabricanți multipli care folosesc ID-uri de produse similare și este nevoie să se urmărească ID produs, ID fabricant, nume produs, preț produs pentru fiecare piesă de echipament din tabela *Products*. Trebuie să vă asigurați că nu sunt permise intrări multiple pentru același produs. Ce instrucțiune ar trebui folosită ?

- A. CREATE TABLE Products (  
productID int PRIMARY KEY,  
manufacturerID int,  
productName varchar(50),  
price decimal(7,2))
- B. CREATE TABLE Products (  
productID int UNIQUE,  
manufacturerID int PRIMARY KEY,  
productName varchar(50),  
price decimal(7,2))
- C. CREATE TABLE Products (  
productID int,  
manufacturerID int,  
productName varchar(50) PRIMARY KEY,  
price decimal(7,2))
- D. CREATE TABLE Products (  
productID int,  
manufacturerID int,  
productName varchar(50),  
price decimal(7,2),  
CONSTRAINT product\_key PRIMARY KEY (productID, manufacturerID))
- E. Nu este posibil de realizat această cerință.

38. Comanda:

```
SELECT DISTINCT codp, denp FROM Produse  
WHERE codp > 200  
ORDER BY cantp
```

simulează implementarea operatorilor relaționali de:

- A. Proiecție și joncțiune
- B. Selecție și intersecție
- C. Selecție și reuniune
- D. Proiecție și selecție
- E. Reuniune și joncțiune

39. Din categoria funcțiilor de grup fac parte:

- A. Funcția COUNT( )
- B. Funcția SUM( )
- C. Funcția ROUND( )
- D. Funcția RTRIM( )
- E. Funcția STDEV( )

40. Dându-se tabelele *Produse*(**codp**, denp, pret ) și *Comenzi*(**nr\_cda**, **codp**, cant, data), instrucțiunile SELECT:

```
SELECT c.*  
FROM comenzi c JOIN produse p  
ON p.codp=c.codp WHERE p.denp='penar'  
și  
SELECT c.nr_cda, c.codp, c.cant, c.data  
FROM comenzi c, produse p  
WHERE p.codp=c.codp  
AND p.denp = 'penar'
```

- A. Sunt diferite ca efect
- B. Sunt identice ca efect
- C. Ar fi identice ca efect dacă în al doilea SELECT s-ar folosi operatorul OR și nu AND
- D. Ar fi identice ca efect dacă în primul SELECT nu s-ar folosi operatorul WHERE

41. Comanda DROP *nume\_tabela* este echivalentă cu comanda:

- A. DELETE FROM *nume\_tabela*
- B. TRUNCATE
- C. ALTER TABLE
- D. ALTER DATABASE
- E. Cu nici una



42. Folosind tabela *cadru\_did*(cod, nume, graddid, data\_ang, sal\_neg, sal\_merit), creată și populată cu date, să se specifice care dintre exemplele următoare sunt corecte:
- A. **SELECT SUM(sal\_neg) FROM cadru\_did**
  - B. SELECT graddid, SUM(sal\_neg) FROM cadru\_did
  - C. **SELECT graddid, SUM(sal\_neg) FROM cadru\_did GROUP BY graddid**
  - D. SELECT graddid, SUM(sal\_neg), COUNT(graddid) FROM cadru\_did  
GROUP BY data\_ang
  - E. SELECT graddid, SUM(sal\_neg) FROM cadru\_did  
WHERE SUM(sal\_neg) > 10000
43. Un utilizator a șters accidental o linie din tabela *Produse*, care conține o coloană **IDENTITY** numită *id*. Când încercați să reinserați aceleași date care au fost în linia ștersă, primiți un mesaj de eroare. Ce ar trebui să faceți?
- A. Puneți pe *on* proprietatea **IDENTITY\_INSERT** folosind SQL Server Management Studio (SSMS) și apoi inserați linia.
  - B. **Executați instrucțiunea SET IDENTITY\_INSERT Produse ON și apoi inserați linia.**
  - C. Ștergeți proprietatea **IDENTITY** pentru coloana *id* folosind SQL Server Management Studio (SSMS) și apoi inserați linia.
  - D. Reinițializați valoarea *id* cu cea a liniei șterse și apoi inserați linia.
44. Scrieți o instrucțiune **SELECT** care extrage date din două tabele. Trebuie să returnați toate liniile din prima tabelă și numai liniile care îndeplinesc condiția din **JOIN** din a doua tabelă. Ce tip de **JOIN** ar trebui să folosiți?
- A. **LEFT**
  - B. **RIGHT**
  - C. **INNER**
  - D. **FULL**
45. Compania dumneavoastră are un număr de mașini care sunt folosite de angajați pentru o perioadă scurtă de timp. În acest scop aveți o tabelă numită *Cars*, care păstrează informații despre mașini și o tabelă numită *CarAssignments*, care stochează mașinile asigurate fiecărui angajat. Trebuie să creați o constrângere care să vă asigure că toate valorile atribuite coloanei *car\_id* din tabela *CarAssignments* corespund unei valori a coloanei *id* din tabela *Cars*. De asemenea, trebuie să vă asigurați că, dacă se șterge o linie din tabela *Cars*, se vor șterge și liniile din tabela *CarAssignments* care o referențiază. Ce instrucțiune ar trebui folosită?
- A. **ALTER TABLE CarAssignments  
ADD CONSTRAINT FK\_car\_id FOREIGN KEY (car\_id)  
REFERENCES Cars (id) ON DELETE CASCADE;**
  - B. ALTER TABLE CarAssignments  
ADD CONSTRAINT FK\_car\_id FOREIGN KEY (car\_id)  
REFERENCES Cars (id) ON DELETE SET NULL;
  - C. ALTER TABLE CarAssignments  
ADD CONSTRAINT FK\_car\_id FOREIGN KEY (car\_id)  
REFERENCES Cars (id);
  - D. ALTER TABLE CarAssignments

```
ADD CONSTRAINT FK_car_id FOREIGN KEY (car_id)
REFERENCES Cars (id) ON DELETE NO ACTION;
```

46. Compania dumneavoastră folosește tabela *WebSites* pentru a urmări informația despre traficul zilnic al mai multor Web sites. Tabela include și următoarele coloane:

- *id*, care stochează ID Web site;

- *hits*, care păstrează numărul de accesări ale site-ului din acea zi.

Trebuie să afișați numărul mediu de accesări zilnice pentru fiecare Web site.

Ce instrucțiune ar trebui să folosiți?

A. `SELECT SUM(hits)/COUNT(id)`

`FROM WebStats`

`GROUP BY id;`

B. `SELECT COUNT(*), AVG(hits)`

`FROM WebStats`

`GROUP BY hits;`

C. `SELECT id, AVG(hits)`

`FROM WebStats`

`GROUP BY id;`

D. `SELECT AVG(hits)`

`FROM WebStats;`

47. Ați creat tabela *Angajati* cu instrucțiunea:

```
CREATE TABLE Angajati (nume varchar(20), departament varchar(20));
```

Se vor adăuga, apoi, patru persoane:

- "Angajat unu" în departamentul Sales;

- "Angajat doi" în departamentul Sales;

- "Angajat trei" în departamentul Marketing;

- "Angajat patru" în departamentul HR.

Trebuie să adăugați acești angajați tablei *Angajati* folosind cea mai mică secvență de cod. Care interogare ar trebui executată în acest scop?

A. `INSERT INTO Angajati (nume, departament)`

`VALUES (' Angajat unu', 'Sales');`

`INSERT INTO Angajati (nume, departament)`

`VALUES (' Angajat doi', 'Sales');`

`INSERT INTO Angajati (nume, departament)`

`VALUES (' Angajat trei', 'Marketing');`

`INSERT INTO Angajati (nume, departament)`

`VALUES (' Angajat patru', 'HR');`

B. `INSERT INTO Angajati (nume, departament)`

`VALUES (' Angajat unu', 'Sales'), (' Angajat doi', 'Sales'),`

`(' Angajat trei', 'Marketing'), (' Angajat patru', 'HR');`

C. `INSERT INTO Angajati`

`VALUES (' Angajat unu', 'Sales');`

`INSERT INTO Angajati`

`VALUES (' Angajat doi', 'Sales');`

`INSERT INTO Angajati`

```
VALUES (' Angajat trei', 'Marketing');  
INSERT INTO Angajati  
VALUES (' Angajat patru', 'HR');
```

**D. INSERT INTO Angajati**

```
VALUES (' Angajat unu', 'Sales'), (' Angajat doi', 'Sales'),  
(' Angajat trei', 'Marketing'), (' Angajat patru', 'HR');
```

48. Aveți o tabelă *Employees*, cu angajații unei firme, care include o coloană de tip `nchar(30)` numită `NumeAng`. Coloana `NumeAng` conține prenumele și numele unui angajat, separate de caracterul underscore (`_`). Trebuie să căutați în tabelă toți angajații al căror nume de familie este `Popa`. Ce expresie ar trebui să folosiți în clauza `WHERE` a instrucțiunii `SELECT`?

- A. `RTRIM(NumeAng) LIKE '%_Popa'`
- B. `NumeAng LIKE '%!_Popa' ESCAPE '!'`
- C. **`RTRIM(NumeAng) LIKE '%!_Popa' ESCAPE '!'`**
- D. `NumeAng LIKE '%Popa%'`

49. Ați creat tabela *Angajati*, în care păstrați informațiile despre angajații firmei unde lucrați, cu instrucțiunea:

```
CREATE TABLE Angajati ( nume varchar(50), telefon char(10), email varchar(20)).
```

Trebuie să vă asigurați că angajații vor putea fi introduși în tabelă cu valoarea `NULL` în coloana `nume`, dar fără valori de tip `NULL` în coloanele `telefon` și `email` (simultan). Cum ar trebui să modificați coloanele tabelii?

- A. Adăugați o constrângere `CHECK` (`telefon IS NOT NULL OR email IS NOT NULL`) pe coloana `telefon`.
- B. Adăugați `NOT NULL` la coloanele `telefon` și `email`.
- C. Adăugați `NULL` la coloana `nume`.
- D. **Add constrângerea `CHECK(telefon IS NOT NULL OR email IS NOT NULL)` pe tabelă.**

50. Ați creat următoarea tabelă cu instrucțiunea:

```
CREATE TABLE dbo.Products (ID bigint IDENTITY(1,1) NOT NULL,  
Name nvarchar(100) NOT NULL, FirstProduction timestamp NOT NULL,  
Cost decimal(12, 4) NOT NULL, SalePrice decimal(12,4) NOT NULL,  
Volume bigint NOT NULL, Margin AS (SalePrice - Cost) / Cost);
```

Trebuie să modificați structura tabelii pentru a utiliza cel mai mic spațiu de stocare. Ce instrucțiuni ar trebui să folosiți?

- A. `ALTER TABLE dbo.Products ALTER COLUMN Margin AS (SalePrice - Cost);`
- B. **`ALTER TABLE dbo.Products REBUILD WITH (DATA_COMPRESSION = ROW);`**
- C. `ALTER TABLE dbo.Products ALTER COLUMN Name ADD SPARSE;`
- D. **`ALTER TABLE dbo.Products ALTER COLUMN Volume int;`**
- E. `ALTER TABLE dbo.Products ALTER COLUMN Cost decimal(10,2);`

## Subiecte SGBD

1. Un SGBD distribuit este:

- A. O mulțime de SGBD-uri locale, independente, ce se execută în același timp
- B. Un software ce permite gestiunea unei BD distribuite și face distribuția transparentă utilizatorilor
- C. O mulțime de aplicații locale și globale
- D. O mulțime de aplicații globale
- E. O mulțime de aplicații locale

2. O tranzacție este:

- A. O unitate de program ce realizează citiri și scrieri
- B. O mulțime finită și ordonată de operații de citire, scriere și terminare
- C. O mulțime finită de operații coerente, executate de un SGBD, care manipulează datele unei baze de date
- D. O mulțime de instrucțiuni executabile dintr-un limbaj gazdă
- E. O serie finită de operații de modificare a datelor dintr-o bază de date ORACLE

3. Proprietatea de consistență a tranzacției constă din:

- A. Eliminarea redundanțelor
- B. Izolarea tranzacțiilor în execuție
- C. Transformarea unei stări consistente într-o stare consistentă
- D. Transparența
- E. Durabilitatea

4. O bază de date distribuită se găsește implementată numai:

- A. Pe un calculator performant
- B. Pe mai multe calculatoare izolate, conectate la Internet
- C. Pe un calculator client
- D. Într-o rețea de calculatoare
- E. Pe un laptop

5. Un SGBD paralel este:

- A. O mulțime de SGBD locale independente în execuție
- B. Un SGBD ce execută în paralel operații de intrare /ieșire dintr-o bază de date
- C. Un program ce operează pe baza algoritmilor paraleli
- D. Un SGBD ce se execută pe calculatoare cu mai multe procesoare și medii de stocare, proiectate să execute o mulțime de operații în paralel, pentru o mare performanță
- E. Niciuna dintre acestea

6. Se dă tabelul

*STUDENT*(CNP int, Nume varchar(30), Grupa nchar(10)), care are zece înregistrări introduse și următoarea secvență ce reprezintă o tranzacție:

BEGIN TRAN

CREATE table STUDENT\_NOU ( CNP int, Nume VARCHAR(30))

INSERT INTO STUDENT\_NOU SELECT CNP, Nume from STUDENT

Save TRAN s1

UPDATE STUDENT\_NOU SET Nume = UPPER(Nume)

Save TRAN s2

DELETE FROM STUDENT\_NOU

Rollback TRAN s2

DELETE FROM STUDENT\_NOU WHERE CNP=100

UPDATE STUDENT\_NOU SET Nume = 'Ionescu'

Rollback TRAN s1

UPDATE STUDENT\_NOU SET Nume = 'Ionescu' WHERE CNP=100

Rollback

Care este rezultatul acestei tranzacții pentru tabela STUDENT\_NOU?

A. Avem un student cu numele 'Ionescu'

**B. Nu mai există tabela STUDENT\_NOU**

C. Ultimul UPDATE eșuează deoarece studentul cu CNP-ul 100 a fost șters

D. Tranzacția eșuează deoarece avem mai multe puncte de reluare

7. Când creai o bază de date Microsoft Sql Server, care sunt fișierele care trebuie create obligatoriu?

A. Un fișier de date primar

**B. Un fișier de date primar și un fișier de log**

C. Un fișier de date primar, un fișier de date secundar și un fișier de log

D. Un fișier de log primar

8. Când se determină accesul utilizator la obiecte particulare ale unei baze de date, cum se aplică permisiunile?

A. User, fixed role, group role, fixed server

B. User, group role, fixed role, fixed server

**C. Permisunile sunt cumulative exceptând DENY**

D. Permisunile se scad exceptând permisiunile user specificate

9. Care dintre următoarele afirmații nu este adevărată despre bazele de date Microsoft SQL Server?

A. Bazele de date sunt grupate în "filegroups"

**B. Bazele de date trebuie să fie în același "filegroup"**

C. Bazele de date pot crește automat cu un procent

D. Bazele de date pot avea dimensiuni restricționate

10. Care grup de instrucțiuni se poate utiliza prin acordarea permisiunilor obiect?

- A. SELECT, UPDATE, INSERT, CREATE
- B. SELECT, EXECUTE, DELETE, INDEX
- C. SELECT, UPDATE, INSERT, DELETE
- D. SELECT, EXECUTE, DELETE, CREATE

11. Ați planificat ca Microsoft SQL Server 2000 să facă backups ale unei baze de date astfel:

full backup la ora 2:00 a.m.; differential backups din 4 în 4 ore; transaction log backups la fiecare 30'. Sistemul cade la 11:24 a.m. După ce se face un full backup restore, câte backups transaction logs și diferențiale (numărul minim) trebuie restaurate pentru a avea pierderi minime în baza de date?

- A. 1 differential, 2 log
- B. 1 differential, 10 log
- C. 2 differential, 2 log
- D. 2 differential, 1 log

12. Pentru a preveni problemele ținând de proprietarul obiectelor unei baze de date Microsoft SQL Server, ce rol ar trebui să fie asignat unui developer?

- A. Db-owner
- B. Db-creator
- C. DBA
- D. Db-manager

13. Rolul Public este un rol special fixat la nivelul unei baze de date Sql Server. Care dintre următoarele afirmații sunt adevărate despre el ?

- A. Nu poate fi șters
- B. Poate fi șters
- C. E conținut în orice bază de date utilizator, inclusiv master, msdb, tempdb, model
- D. E conținut numai în bazele de date utilizator
- E. Nu poate avea utilizatori, grupuri sau roluri asignate
- F. Păstrează toate permisiunile implicite pentru utilizatorii unei baze de date

14. Care dintre afirmațiile următoare sunt adevărate?

- A. Numai un index cluster poate fi creat pentru un tabel
- B. Întotdeauna creați indecșii non-cluster înaintea creării oricărui index cluster
- C. Indecșii cluster sunt mai mari decât cei non-cluster
- D. Paginile frunză ale indexului cluster sunt paginile de date ale tabelului

15. Trebuie să modificați un tabel din baza de date pe care ați creat-o; mai mulți utilizatori au primit permisiunile obiect SELECT, INSERT pentru acel tabel. Ce variantă ar trebui să executați pentru a modifica tabelul, fără a afecta permisiunile acordate?

- A. DROP TABLE urmat de CREATE TABLE
- B. CREATE TABLE, cu noua configurație
- C. ALTER TABLE

D. MOD TABLE

E. EXEC sp\_help nume\_tabela

16. Când ștergeți un tabel, ce obiecte din baza de date se șterg de asemenea?

A. Vederile care-l au ca tabel de bază

B. Procedurile stocate care-l utilizează

C. Declanșatorii creați pentru el

D. Funcțiile care-l utilizează

17. Care sunt caracteristicile unei chei primare?

A. Forțează integritatea referențială pentru o tabelă

B. Se crează cu coloanele existente în tabel sau cu o coloană creată special în acest scop

C. Permite valori de tip null

D. Server-ul de baze de date permite numai o cheie primară pentru un tabel

E. La crearea ei, serverul construiește automat o constrângere de domeniu

18. Care dintre următoarele afirmații sunt adevărate despre constrângeri?

A. Constrângerile la nivel de coloană se aplică pe anumite coloane sau combinații de coloane

B. Dacă nu se specifică un nume pentru constrângere, server-ul de baze de date oferă unul

C. Indecșii creați de constrângerile PRIMARY KEY și UNIQUE KEY pot fi șterși folosind instrucțiunea DROP INDEX

D. Constrângerile pot fi adăugate sau șterse dintr-o tabelă fără a afecta structura tablei

19. Puteți folosi o procedură stocată pentru a extrage informații din tabele de bază la care nu aveți permisiuni de acces?

A. Nu, pentru că nu aveți permisiunile cerute pe tabele

B. Nu, dacă nu sunteți membri sysadmin sau db-owner

C. Da, dacă aveți permisiunea EXECUTE pentru procedura stocată

D. Da, dacă ați creat declanșatori pentru fiecare tabel

20. Dacă utilizatorul *student* crează o tabelă, cine poate interoga tabela?

A. Numai utilizatorul *student*

B. Numai administratorul bazei de date

C. Utilizatorii care au drept de SELECT pe tabelă

D. Toți utilizatorii

E. Utilizatorul *student*

21. Să se aleagă răspunsurile corecte:

A. Când o tabelă e ștearsă, toți indecșii asociați sunt șterși

B. Întotdeauna se construiește un index pe coloanele care se actualizează frecvent

C. Se poate crea un index ce conține câmpuri din mai multe tabele

D. Pentru a obține date ordonate dintr-o tabelă se pot folosi indecși

E. Se recomandă crearea unui index pe coloanele folosite în joncțiuni

22. Comanda DROP VIEW *vedere* are ca efect:

- A. Șterge înregistrările din tabela virtuală
- B. Șterge înregistrările din tabela de bază
- C. Șterge tabela virtuală
- D. Șterge tabela de bază
- E. Șterge tabela virtuală și tabela de bază corespunzătoare

23. Serverul Oracle/Microsoft de baze de date crează automat un index când:

- A. Se definește o constrângere de tip PRIMARY KEY
- B. Se definește o constrângere de tip NOT NULL
- C. Se definește o constrângere de tip UNIQUE
- D. Se definește o constrângere de tip FOREIGN KEY
- E. Se definește o constrângere de tip CHECK

24. Se va crea un index pe o coloană când:

- A. Coloana se utilizează frecvent în clauza WHERE sau într-un JOIN
- B. Tabela este de dimensiuni mari
- C. Tabela este de dimensiuni mici
- D. Coloana conține un număr majoritar de valori distincte
- E. Tabela este actualizată frecvent

25. Secvența de comenzi:

```
SET AUTOCOMMIT OFF  
DELETE FROM stoc  
ROLLBACK
```

are ca efect:

- A. Nu execută nici o ștergere
- B. Șterge toate înregistrările din tabela stoc
- C. Șterge doar înregistrarea curentă din tabela stoc
- D. Șterge fizic toate înregistrările din tabela stoc
- E. Șterge logic toate înregistrările din tabela stoc

26. Care dintre următoarele instrucțiuni sunt necesare pentru a defini începutul și sfârșitul unei tranzații explicite în Microsoft Transact-SQL ?

- A. OPEN TRANSACTION
- B. BEGIN TRANSACTION
- C. INSERT TRANSACTION
- D. END TRANSACTION
- E. ROLLOVER TRANSACTION
- F. COMMIT TRANSACTION

27. Ce problemă de concurență apare când o tranzație citește datele necomise de o altă tranzație aflată în derulare?

- A. Lost update
- B. Dirty Read
- C. Nonrepeatable Read



D. Phantom Read

28. Care dintre următoarele constrângeri pot fi făcute disable?

A. DEFAULT

B. CHECK

C. PRIMARY KEY

D. UNIQUE

E. FOREIGN KEY

29. Trebuie să modificați o procedură stocată și mai mulți utilizatori au primit permisiunea de execuție a ei. Ce instrucțiune ar trebui să utilizați pentru a o modifica, fără a afecta permisiunile existente?

A. DROP PROC

B. CREATE PROC

C. ALTER PROC

D. MOD PROC

E. EXEC PROC

30. Care sunt tipurile principale de integritate a datelor implementate de constrângerile din serverele relaționale de baze de date ?

A. Integritatea de domeniu

B. Integritatea de atribut

C. Integritatea entității

D. Integritatea referențială

E. Integritatea tipurilor de date

31. Ce caracteristică este utilizată pentru a preveni conflictele de actualizare, astfel încât utilizatorii să nu poată citi sau modifica datele când alți utilizatori le modifică?

A. Tranzacțiile

B. Blocările

C. Interogările de tip SQL

D. Permiuniunile de tip UPDATE

E. Permiuniunile de tip SELECT

32. Ați creat o vedere folosind comanda:

```
CREATE VIEW dbo.Employee
```

```
WITH ENCRYPTION
```

```
AS SELECT Name FROM dbo.HumanResources
```

```
WHERE isEmployee = 1
```

```
WITH CHECK OPTION
```

Trebuie să împiedicați alți utilizatori să copieze vederea. Ce instrucțiune ar trebui să folosiți?

A. ALTER VIEW dbo.Employee

WITH ENCRYPTION

AS SELECT Name FROM dbo.HumanResources

WHERE isEmployee = 1

- B. DENY ALL ON dbo.Employee TO public
- C. ALTER VIEW dbo.Employee  
AS SELECT Name FROM dbo.HumanResources  
WHERE isEmployee = 1  
WITH CHECK OPTION
- D. ALTER VIEW dbo.Employee  
WITH SCHEMABINDING  
AS SELECT Name FROM dbo.HumanResources  
WHERE isEmployee = 1
- E. Nu exista o asemenea instructiune

33. Ați executat trei proceduri stocate folosind următoarele instrucțiuni SQL:

```
BEGIN TRANSACTION  
EXEC employeeStats  
SAVE TRANSACTION SV_employee  
EXEC productStats  
SAVE TRANSACTION SV_product  
EXEC webStats
```

A apărut o eroare în procedura stocată productStats și este nevoie să se anuleze modificările făcute de acea procedură.

Ce instrucțiune ar trebui să executați?

- A. ROLLBACK TRANSACTION
- B. ROLLBACK TRANSACTION SV\_employee
- C. ROLLBACK WORK
- D. ROLLBACK  
ROLLBACK
- E. ROLLBACK TRANSACTION SV\_product

34. Folosiți tabela *Products* pentru a stoca informații despre produse. Printre alte informații, tabela conține numele și prețul produselor. Creșteți prețul produselor astfel:

- prețul produselor mai mic de 100\$ va fi crescut cu 10% ;
- prețul produselor cuprins în intervalul 100\$ și 1000\$ va fi crescut cu 5%;
- prețul produselor mai mare de 1000\$ va fi crescut cu 1%.

Trebuie să faceți aceste modificări pentru toate produsele din tabela *Products* și să afișați lista cu numele produselor, prețul original și prețul crescut. Ce instrucțiune ar trebui folosită ?

- A. DECLARE @t1 table  
(name varchar(30),  
oldprice decimal(7,2),  
newprice decimal(7,2));  
UPDATE Products  
SET price =  
CASE  
WHEN price < 100.00 THEN price\*1.10  
WHEN price BETWEEN 100.00 AND 1000.00 THEN price\*1.05  
ELSE price\*1.01

END  
OUTPUT DELETED.name, DELETED.price, INSERTED.price INTO @t1

B. UPDATE Products

SET price =  
CASE  
WHEN price < 100.00 THEN price\*1.10  
WHEN price BETWEEN 100.00 AND 1000.00 THEN price\*1.05  
ELSE price\*1.01  
END

OUTPUT DELETED.name, DELETED.price

C. UPDATE Products

SET price =  
CASE  
WHEN price < 100.00 THEN price\*1.10  
WHEN price BETWEEN 100.00 AND 1000.00 THEN price\*1.05  
ELSE price\*1.01  
END

OUTPUT DELETED.name, DELETED.price, INSERTED.price

D. UPDATE Products

SET price =  
CASE  
WHEN price < 100.00 THEN price\*1.10  
WHEN price BETWEEN 100.00 AND 1000.00 THEN price\*1.05  
ELSE price\*1.01  
END

OUTPUT DELETED.\*, INSERTED.\*

E. UPDATE Products

SET price =  
CASE  
WHEN price < 100.00 THEN price\*1.10  
WHEN price BETWEEN 100.00 AND 1000.00 THEN price\*1.05  
ELSE price\*1.01  
END

OUTPUT DELETED.name, DELETED.price, INSERTED.\*

35. O tabelă virtuală:

- A. Poate fi construită dintr-o tabelă de bază
- B. Nu poate fi construită din mai multe tabele de bază legate între ele (cu UNION)
- C. Poate conține funcții de grup
- D. Nu poate fi read only
- E. Poate conține o restricție de validare

36. Să se aleagă afirmațiile corecte:

- A. Când se șterge o tabelă, indecșii asociați rămân până se șterge și baza de date
- B. Pentru a obține date ordonate dintr-o tabelă se pot folosi indecși
- C. Întotdeauna se construiește un index pe attributele care se actualizează în mod curent
- D. Se poate construi un index care conține câmpuri din mai multe tabele
- E. Se recomandă crearea unui index pe attributele folosite în joncțiuni

37. Să se aleagă informațiile corecte:

- A. O vedere poate fi creată numai în baza de date curentă
- B. În comanda CREATE VIEW nu se poate folosi operatorul UNION
- C. În comanda CREATE VIEW nu se poate folosi clauza ORDER BY decât cu clauza TOP
- D. În comanda CREATE VIEW se poate folosi instrucțiunea SELECT
- E. În comanda CREATE VIEW se poate folosi clauza SET

38. Se poate folosi același nume pentru mai multe tabele stocate pe un server de baze de date Microsoft SQL Server ?

- A. Nu
- B. Da, dacă nu aparțin aceleleași baze de date
- C. Da, dacă aparțin aceleleași scheme
- D. Da, dacă nu aparțin aceleleași scheme
- E. Da, dacă nu aparțin aceleleași instanțe

39. Dacă utilizatorul "stud" crează o vedere, cine poate face interogări pe acea vedere?

- A. Administratorul bazei de date
- B. Utilizatorii care au permisiunea SELECT pe tabela din care extrage vederea datele
- C. Utilizatorul "stud"
- D. Numai utilizatorul "stud"
- E. Toți utilizatorii care au permisiunea SELECT pe vedere

40. Un index Microsoft SQL/Oracle Server poate fi șters de:

- A. Proprietarul tabelii
- B. De orice utilizator care are dreptul DROP ANY INDEX
- C. Numai de proprietarul tabelii
- D. De administratorul bazei de date
- E. Numai de administratorul bazei de date

41. Un administrator de bază de date poate:

- A. Crea un utilizator
- B. Șterge un utilizator
- C. Șterge orice tabelă sistem
- D. Șterge orice index creat pe o tabelă sistem
- E. Să salveze orice bază de date utilizator

42. Secvența de comenzi Microsoft SQL Server:

```
SET IMPLICIT_TRANSACTIONS OFF
BEGIN TRANSACTION
DELETE FROM Tabel_sursa
ROLLBACK
```

are ca efect:

- A. Șterge toate înregistrările din tabela "Tabel\_sursa"
- B. Șterge doar înregistrarea curentă din "Tabel\_sursa"
- C. Nu efectuează nici o ștergere
- D. Șterge toți indecșii asociați tablei "Tabel\_sursa"
- E. Șterge logic toate înregistrările din tabela "Tabel\_sursa"

43. Compania dumneavoastră utilizează o procedură stocată numită *webAnalysis* pentru a analiza informația de pe un Web site. Procedura întoarce 1 dacă analiza a avut loc cu succes și 0 dacă a existat o problemă. Dumneavoastră creați o interogare care execută un grup de operații de întreținere zilnice, incluzând procedura *webAnalysis*. Trebuie să comiteți modificările făcute de procedură, dacă analiza a înregistrat succes, fără să afectați execuția altor task-uri executate de interogare. Care set de instrucțiuni ar trebui să folosiți?

- A. 

```
BEGIN TRANSACTION;
DECLARE @success int;
EXEC @success = webAnalysis;
IF @success = 0
    ROLLBACK;
ELSE
    COMMIT;
```
- B. 

```
BEGIN TRANSACTION;
DECLARE @success int;
EXEC @success = webAnalysis;
IF @success = 0
    ROLLBACK;
```
- C. 

```
BEGIN TRANSACTION;
DECLARE @success int;
EXEC @success = webAnalysis;
COMMIT;
IF @success = 0
    ROLLBACK;
```
- D. 

```
DECLARE @success int;
EXEC @success = webAnalysis;
IF @success = 0
    ROLLBACK;
ELSE
    COMMIT;
```

44. Inserați date într-o tabelă care folosește un trigger INSTEAD OF pentru a le valida, înainte de a le adăuga; dacă datele nu sunt corecte, ele nu sunt inserate în tabelă. Trebuie să scrieți o instrucțiune INSERT și să returnați o listă a liniilor care au fost validate și adăugate în tabelă. Ce ar trebui să faceți?

- A. Creați o variabilă de tip tabelă pentru a stoca datele inserate. Folosiți clauza OUTPUT INTO în instrucțiunea INSERT pentru a popula variabila tabelă cu liniile inserate. Executați instrucțiunea SELECT pe variabila tabelă pentru a returna liniile inserate.
- B. Creați o variabilă de tip tabelă pentru a stoca datele inserate. Folosiți clauza OUTPUT INTO în instrucțiunea INSERT pentru a popula variabila tabelă cu liniile inserate. Executați instrucțiunea SELECT care întoarce toate liniile din tabelă și variabila tabelă.
- C. Folosiți clauza OUTPUT în instrucțiunea INSERT pentru a returna liniile inserate.
- D. Creați o variabilă de tip tabelă și o populați cu conținutul inițial al tablei. Inserați datele în tabelă. Executați o instrucțiune SELECT care extrage toate liniile din tabelă, dar care nu sunt în variabila tabelă.

45. Tabela *Employees* include următoarele coloane:

- *id* int UNIQUE;
- *name* nvarchar(50).

Trebuie să creați o procedură stocată care să permită utilizatorilor să transmită ID-ul unui angajat și să extragă numele angajatului pe care sa-l memoreze într-o variabilă *nvarchar(50)*. Ce instrucțiune ar trebui să utilizați?

- A. CREATE PROCEDURE getName  
@ID int  
AS  
DECLARE @outputName nvarchar(50);  
SELECT @outputName = name FROM Employees WHERE id = @ID;  
RETURN @outputName;
- B. CREATE PROCEDURE getName  
@ID int,  
@Name nvarchar(50) OUTPUT  
AS  
SELECT @Name = name FROM Employees WHERE id = @ID;
- C. CREATE PROCEDURE getName  
@ID int,  
@Name nvarchar(50)  
AS  
SELECT @Name = name FROM Employees WHERE id = @ID;
- D. CREATE PROCEDURE getName  
@ID int  
AS  
SELECT name FROM Employees WHERE id = @ID;

46. Baza de date folosită de dumneavoastră conține o tabelă numită *Employees*, care are o coloană de tip *nvarchar(MAX)* numită *lastName*. Aveți deja un index clustered numit *id\_index* pe coloana *id* a tablei. Unul dintre utilizatorii dumneavoastră reclamă timpii de interogare lungi obținuți când folosește coloana *lastName* în clauza WHERE a instrucțiunii SELECT. Ce instrucțiune ar trebui să folosiți?

- A. ALTER INDEX id\_index  
ON Employees (id, lastName);
- B. CREATE INDEX id\_name\_index  
ON Employees (id)  
INCLUDE (lastName);
- C. CREATE CLUSTERED INDEX name\_index  
ON Employees (lastName);
- D. CREATE INDEX name\_index  
ON Employees (lastName);

47. O bază de date folosită de dumneavoastră include o tabelă numită *Contractors*. Aplicațiile folosesc frecvent următoarea instrucțiune pentru a accesa înregistrările actualizate după 1 ianuarie 2014:

```
SELECT id, name FROM Contractors  
WHERE expertise = @searchWord  
AND lastUpdated > '20140101';
```

Trebuie să reduceți timpul necesar execuției acestei instrucțiuni. Care este soluția optimă pentru a obține acest deziderat?

- A. CREATE INDEX expertise\_index  
ON Contractors (lastUpdated, expertise);
- B. CREATE INDEX expertise\_index  
ON Contractors(lastUpdated)  
INCLUDE (expertise);
- C. CREATE INDEX expertise\_index  
ON Contractors (expertise)  
WHERE lastUpdated > '20140101';
- D. CREATE INDEX expertise\_index  
ON Contractors (expertise, lastUpdated);

48. În baza de date folosită de dumneavoastră aveți o tabelă numită *Personal*, al cărei proprietar este *student*. S-a creat o vedere numită *Angajati*, pentru a limita accesul la tabela *Personal*, care are ca proprietar pe *Ana*. Unuia dintre utilizatorii bazei de date i-a fost acordată permisiunea INSERT pe vederea *Angajati* și i-a fost interzisă permisiunea INSERT pe tabela *Personal*; când încearcă să insereze o linie în vederea *Angajati*, instrucțiunea INSERT eșuează. Trebuie să rezolvați această problemă. Ce ar trebui să faceți?

- A. Acordați proprietarului tablei *Personal* permisiunea SELECT pe vederea *Angajati*.
- B. Acordați proprietarului vederii *Angajati* permisiunea INSERT pe tabela *Personal*.
- C. Schimbați proprietarul vederii *Angajati* în *student*.
- D. Acordați permisiunea INSERT pe tabela *Personal* utilizatorului.

49. Ați creat o vedere folosind comanda:

```
CREATE VIEW dbo.AngajatiNoi  
AS SELECT  
    Nume,  
    SalariuLunar,  
    (SalariuLunar * 12) AS SalariuAnual,  
    Data_Ang  
FROM dbo.ResurseUmane  
WHERE Nume IS NOT NULL  
    AND Data_Ang > '1/1/2014'  
WITH CHECK OPTION;
```

Trebuie să inserați o linie folosind această vedere. Ce instrucțiune ar trebui să folosiți?

- A. INSERT INTO dbo. AngajatiNoi (Nume, SalariuAnual, Data\_Ang) VALUES ('Popa Ion', 50000, '3/12/2014');
- B. INSERT INTO dbo. AngajatiNoi (Nume, SalariuLunar, Data\_Ang) VALUES ('Mihnea George', 4000, '5/13/2014');
- C. INSERT INTO dbo. AngajatiNoi (Nume, SalariuLunar, Data\_Ang) VALUES ('Stan Remus', 2500, '11/5/2010');
- D. INSERT INTO dbo. AngajatiNoi (Nume) VALUES ('Tonoiu Petre');

50. Ați creat o tabelă folosind instrucțiunea:

```
CREATE TABLE dbo.Products (ID int IDENTITY(1,1) NOT NULL,  
    Name nvarchar(60) NOT NULL, Cost decimal(10,2),  
    SalePrice decimal(10,2), CurrentStock bigint, NumberSold bigint)
```

Un utilizator necesită următoarele tipuri de acces:

- SELECT pe coloanele Name și SalePrice;
- SELECT pe stocul disponibil(diferența dintre CurrentStock și NumberSold );
- ALTER pe coloanele Name și SalePrice.

Trebuie să acordați permisiuni pentru a satisface aceste cerințe, fără a da permisiuni suplimentare sau a restricționa accesul. Ce ar trebui să faceți?

- A. Acordați utilizatorului permisiunile SELECT și ALTER pe vederea definită de instrucțiunea: CREATE VIEW dbo.CustomerProduct AS SELECT Name, SalePrice, (CurrentStock - NumberSold) AS AvailableStock FROM dbo.Products.
- B. Acordați utilizatorului permisiunea SELECT pe coloanele Name, SalePrice, CurrentStock și NumberSold și permisiunea ALTER pe coloanele Name și SalePrice din tabela *Products*.
- C. Acordați utilizatorului permisiunea ALTER pe vederea definită de instrucțiunea: CREATE VIEW dbo.CustomerProductSelect AS SELECT Name, SalePrice, (CurrentStock - NumberSold) AS AvailableStock FROM dbo.Products.
- D. Creați două proceduri stocate, una care să execute instrucțiunea SELECT din tabelă și alta care să permită ALTER pe coloanele Name și SalePrice, apoi acordați permisiunea EXECUTE pe proceduri.



**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**MODULUL 3**

***Sisteme de operare, rețele de calculatoare și securitatea  
informației***

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Sisteme de operare**

# Sisteme de Operare

Observatie:

Semnul punct si virgula (;) de la sfarsitul variantelor si semnul punct (.) de la ultima varianta nu face parte din comanda.

1. Planificarea Round-Robin:

- a. determină atribuirea unei cuante de timp în care procesului îi este permis să ruleze;
- b. determină suspendarea unui proces într-un moment arbitrar;
- c. determină atribuirea unei priorități fixe pentru execuția unui proces;
- d. determină lansarea în execuție a unui proces dacă nu există altul mai prioritar în execuție.

2. La declanșarea unei întreruperi prima acțiune desfășurată în sistemul de calcul este:

- a. salvarea registrelor procesului întrerupt în tabela de procese;
- b. încărcarea registrului program counter cu conținutul vectorului de întrerupere;
- c. salvarea stării programului întrerupt în stivă;
- d. modificarea stării procesului interrupt.

3. Un proces este:

- a. activitatea unității de prelucrare;
- b. un program compilat;
- c. un program în execuție;
- d. un program link-editat.

4. Registrul "contor program" conține:

- a. adresele operanzilor operației ce trebuie executată;
- b. adresa următoarei instrucțiuni ce trebuie executată;
- c. chiar operanzii operației ce trebuie executată;
- d. numărul instrucțiunii ce trebuie executată.

5. Secțiunile critice sunt:

- a. secvențe de program care trebuie tratate cu mare atenție;
- b. secvențe de tratare a întreruperilor importante;
- c. secvențe din activitatea sistemului de calcul când resursele sunt insuficiente;
- d. secvențe de program unde se accesează date partajate.

6. Planificatorul este componenta sistemului de operare care decide:

- a. care proces trece în starea de execuție și pentru cât timp;
- b. care proces trebuie terminat și care proces trebuie început;
- c. care dispozitiv periferic va fi folosit de către program;
- d. care dintre fișierele deschise trebuie închis la un moment dat.

7. Un sistem de operare asigură îndeplinirea următoarelor funcții:

- a. creșterea performanțelor unității de prelucrare;
- b. un management eficient al resurselor fizice și logice;
- c. asistarea utilizatorului printr-un sistem dezvoltat de comunicație;
- d. protecția informației.

8. Tranziția unui proces din starea “în execuție” în starea “blocat” are loc atunci când:

- a. planificatorul decide suspendarea procesului pentru a lansa un altul mai prioritar;
- b. planificatorul decide blocarea procesului din cauza unei erori;
- c. procesul așteaptă apariția unor date de care depinde continuarea execuției;
- d. procesul nu mai poate continua execuția din cauza unei erori.

9. În sistemele de operare de tip client-server nucleul acestuia realizează:

- a. servirea proceselor server;
- b. servirea proceselor client;
- c. transportul mesajelor de la procesele client către procesele server;
- d. accesul la resursele proceselor client și ale proceselor server.

10. Evacuarea-reîncărcarea proceselor (swapping) este procesul de:

- a. terminare, respectiv începere a proceselor;
- b. transfer al datelor în fișiere pe disc;
- c. lansare în execuție a programelor;
- d. transfer al proceselor între memorie și disc.

11. Tranziția unui proces din starea “în execuție” în starea “gata” are loc atunci când:

- a. planificatorul decide suspendarea procesului pentru că i-a expirat cuanta de timp alocată;
- b. planificatorul, în urma analizei proceselor active stabilește că un proces nu poate continua execuția;
- c. procesul nu mai are date de prelucrat și trebuie să-și termine execuția;
- d. procesul nu mai are date de transferat și poate să-și continue execuția.

12. Într-un sistem de operare de tip client-server procesele server lucrează în modul de lucru:

- a. supervisor;
- b. kernel;
- c. utilizator;
- d. privilegiat.

13. Care din următoarele tipuri nu reprezintă un sistem de fișiere:

- a. Swap;
- b. ext1;
- c. NTFS;
- d. FAT32.

14. În cadrul partiționării unui sistem Linux, reprezintă un tip de partiție?

- a. Partiția logică;
- b. Partiția primară;
- c. Partiția virtuală;
- d. Partiția extinsă.

15. RUID reprezintă:

- a. Userul logat inițial în sistem;
- b. Userul devenit prin rularea comenzii **su**;
- c. RUID nu există. Doar EUID și UID;
- d. UID secundar.

16. EUID reprezintă:

- a. Userul care s-a logat inițial în sistem;
- b. Userul devenit prin rularea comenzii **su**;
- c. UID secundar al userului;
- d. EUID nu există, doar UID.

17. Ce comandă se folosește pentru afișarea userilor logați în sistem?

- a. login;
- b. whosin;
- c. who;
- d. Whoami.

18. Cum se află procesul programului care deschide portul 10000 pentru a asculta conexiuni ce folosesc protocolul tcp?

- a. `fuser -n tcp 10000;`
- b. `grep 10000;`
- c. `kill 10000;`
- d. `fuser -n tcp.`

19. Comanda `su-` are drept scop:

- a. Substituirea userului curent cu root și încărcarea environmentului lui root;
- b. Substituirea userului curent fără root și încărcarea environmentului lui root;
- c. O eroare fiindcă s-a specificat semnul minus dar nu și opțiunea;
- d. Modificarea informațiilor despre un user.

20. Afișarea tuturor grupurilor din care face parte un utilizator numit *user1* se realizează prin comanda:

- a. `groups;`
- b. `show groups;`
- c. `id;`
- d. `show users.`

21. Ce comandă creează un user numit *user1*?

- a. `useradd user1;`
- b. `useradd user1 -c 'user normal' -d /home/user1 -s /bin/bash;`
- c. `useradd -c 'user normal' -d /home/user1 -m -s /bin/bash user1;`
- d. `adduser -group root user1.`

22. Pentru fiecare user se pastrează în fisierul */etc/passwd* următoarele informații:

- a. `username;`
- b. `uid;`
- c. grupurile din care acesta face parte;
- d. `home directory.`

23. Cum ajungem în directorul home al utilizatorului logat?

- a. `cd /;`
- b. `cd ~;`
- c. `cd home;`

d. cd.

24. În Linux în mod standard userii se găsesc în fișierul:

- a. /etc/users;
- b. /etc/shadow;
- c. /etc/system-users;
- d. /etc/passwd.

25. Ce comandă folosim în linux pentru a afla adresa ip a placii de rețea?

- a. ifconfig;
- b. ipconfig;
- c. route;
- d. traceroute.

26. Care sunt firmațiile pentru care echivalența dintre modul simbolic și absolut este corectă?

- a. 775=rwxrwxrwx;
- b. 775=rwxrwxr-x;
- c. 777=rwxrwxrwx;
- d. 664=rw-rw-r—.

27. *Umask* are rolul de a:

- a. Stabili cu ce permisiuni “default” se crează noile fișiere și directoare;
- b. Stabili cu ce permisiuni default se crează fișierele dacă ne referim la files umsk;
- c. Stabili masca sistemului;
- d. Stabili masca userului.

28. Dacă se folosește umask 0002 care dintre următoarele afirmații sunt corecte:

- a. Fișierele se crează “default” cu permisiunile 664;
- b. Fișierele se crează “default” cu permisiunile 440;
- c. Directoarele se crează “default” cu permisiunile rwxrwxr-x;
- d. Directoarele se crează “default” cu permisiunile 755.

29. Cum se închide procesul 2543?

- a. kill -9 2543;
- b. ps – aux | 2543;

- c. `grep 2543;`
- d. `exit 2543.`

30. În ce fișier se scriu comenzile ce se vor executate la pornirea calculatorului după pornirea tuturor proceselor?

- a. `/etc/rc.d/rc.local;`
- b. `crontab;`
- c. `/bin/sh;`
- d. `/bin.`

31. Ce se scrie în prima linie a unui script bash?

- a. `#!/bin/bash;`
- b. `#!/usr/local/perl;`
- c. `#!/sbin/perl;`
- d. `#!/bin.`

32. Ce definește directiva `DirectoryIndex` din fișierul de configurare a serverului *Apache*?

- a. directorul în care trebuie pus fișierul `index.html`;
- b. fișierele pe care le caută în Document Root pentru a le inițializa ca primă pagină a site-ului;
- c. indexul cuvintelor de căutare a paginii web folosind un motor ca `www.google.com`;
- d. directorul în care intră doar root dacă furnizează parola.

33. Cu ce comandă aflăm direct procesul folosit de un utilizator conectat prin *ssh* la server?

- a. `ps -aux | grep pts;`
- b. `ps -aux | who;`
- c. `ps -aux | grep last;`
- d. `ps -aux | whoami.`

34. Cum facem pentru a avea anumite servicii pornite automat la pornirea calculatorului?

- a. folosim comanda *setup*, apoi din meniu alegem *system services* și bifăm acele servicii;
- b. edităm fișierul `/etc/xinetd`;
- c. edităm fișierul `/etc/inet.d/inetd`;
- d. edităm un fișier nou în care enumerăm serviciile.

35. Folosind interpretorul bash se creează următorul script denumit *scr1*:

```
# !/bin/bash
```



echo \$#

Care va fi rezultatul afisat, dacă scriptul se va executa cu comanda: # bash scr1 arg1 arg2 4:

- a. 3;
- b. arg1;
- c. arg2;
- d. 4.

36. Ce rezultat va produce execuția urmatorului script:

```
#!/bin/bash
s=0
for i in `ls -l | awk '{print $5}'`
do
s=`expr $s + $i`
done
echo `expr $s / 1024`
```

- a. scriptul afisează valoarea însumata, exprimată în Kb, a fișierelor din directorul current;
- b. scriptul afisează o lista cu dimensiunea fișierelor, în octeti din directorul curent;
- c. scriptul afisează valoarea insumata, exprimată înbiti, a fișierelor din directorul curent;
- d. scriptul afisează o listă cu dimensiunea fișierelor din directorul curent exprimată în biți.

37. Se presupune că ați editat un script cu numele /root/scr1. Care este comanda ce trebuie introdusă pentru a face scriptul executabil?

- a . chmod u+x /root/scr1;
- b . chmod 744 /root/scr1;
- c . chmod 666 /root/scr1;
- d . chmod u-wx /root/scr1.

38. Care este rezultatul afișat în urma execuției comenzii:

```
ifconfig | grep "inet addr" | grep -v "127.0.0.1" | awk '{print $2;}' | awk -F':' '{print $2;}'
```

- a . toate adresele ip mai puțin adresa interfeței loopback;
- b . adresa ip a interfeței eth0;
- c . adresa ip a interfeței loopback;
- d . toate adresele ipv6.

39. Care este rezultatul afișat în urma execuției:

*init 3*

- a . intrarea în nivelul 3, respectiv modul text, cu acces network;
- b . intrarea în nivelul 3, respectiv modul grafic cu acces network;
- c . crearea a 3 procese în memorie, subordonate procesului parinte *init*.

40. Care dintre următoarele comenzi are ca rezultat crearea unui fișier *file2* care este identic cu fișierul *file1*, ignorând liniile goale?

- a . `grep -v "^$" file1 > file2;`
- d . `sed -e '/^$/d' file1 > file2;`
- b . `grep -v "#^" file1 > file2;`
- c . `grep "^$" file1 > file2.`

41. Cu ce comandă se poate obține lista rutelor cunoscute de sistem?

- a . `route;`
- b . `netstat -r;`
- c . `netstat -n;`
- d . `ifconfig.`

42. Care este rezultatul afișat în urma execuției următoarei comenzi:

*ps -ef --no-heading | awk '{print \$1}' | sort | uniq*

- a . numele tuturor userilor, fara valori duplicate, sub care sunt rulate procesele curente;
- b . lista id-urilor aparținând proceselor curente, fără valori duplicate;
- c . numarul de useri, fără valori duplicate, sub care sunt rulate procesele curente;
- d . lista numelor proceselor curente, fără valori duplicate, sortate alfabetic.

43. Comanda *echo \$* are următoarea semnificație?

- a . se afișează codul de eroare al ultimei comenzi;
- b . se afișează numarul de parametri ai scriptului curent;
- c . se afișează numele scriptului curent;
- d . se afișează variabila de sistem \$HOME.

44. Care este comanda folosită pentru a porni serviciul *network* în sistemul de operare Linux ?

- a . `service network start;`
- b . `/etc/init.d/network start;`
- c . `chkconfig network start;`
- d . `sc start network.`

45. Ce nu se poate obține folosind comanda *uname*?

- a . numarul de procesoare;
- b . totalul de memorie disponibilă;
- c . versiunea kernel-ului;
- d . numele sistem;
- e . denumirea kernel-ului.

46. Pentru a funcționa rezoluția de nume, ce fisier trebuie să existe și să conțină una sau mai multe adrese ale serverelor DNS?

- a . */etc/resolv.conf*;
- b . */etc/sysconfig/network*;
- c . */etc/sysconfig/network-scripts/network-functions*;
- d . */etc/hosts*.

47. Care sunt informațiile care nu se regăsesc în fișierul */etc/passwd*?

- a. Hash-urile parolelor utilizatorilor;
- b. Directorul home al utilizatorului curent;
- c. Interpretorul de comenzi (shell-ul);
- d. Grupurile secundare ale utilizatorului curent.

48. Ce fișier este folosit pentru montarea automată a unui sistem de fișiere în sistemul de operare Linux?

- a. */etc/fstab*;
- b. */dev/sda1*;
- c. */mnt/hda6*;
- d. */etc/fsck*

49. Care este rezultatul afișat în urma execuției următoarei comenzii:

```
cat /etc/passwd | grep nologin | cut -d: -f1
```

- a . numele utilizatorilor care nu se pot loga în sistem;
- b . numele utilizatorilor al căror cont a fost dezactivat;
- c . numele grupului utilizatorilor al căror cont a fost dezactivat;
- d . numele grupului utilizatorilor care nu se pot loga în sistem.

50. Care este rezultatul afișat în urma execuției următoarei comenzii:

```
find -type f -atime -7
```

- a . căutarea în directorul curent a fișierelor accesate în ultimele 7 zile;
- b . căutarea în directorul rădăcină a fișierelor accesate în ultimele 7 ore;
- c . căutarea în directorul rădăcină a fișierelor create în ultimele 7 ore;
- d . căutarea în directorul curent a fișierelor create în ultimele 7 zile.

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Retele de calculatoare**

**&**

**Disciplina Administrarea rețelelor de calculatoare**

# MODELE ÎNTREBĂRI EXAMEN LICENȚĂ 2015

## Disciplina Rețele de calculatoare

&

## Disciplina Administrarea rețelelor de calculatoare

1. Ce algoritmi se pot folosi pentru a calcula distanța minimă dintre noduri într-o rețea?
  - a) algoritmul inundații
  - b) despicarea orizontului
  - c) algoritmul lui Dijkstra
  - d) algoritmi pe bază de flux
2. Rutarea este:
  - a) procesul de transfer al datelor prin două sau mai multe rețele, de la o sursă către o destinație
  - b) procesul de transfer al datelor într-un segment de rețea de la o sursă la o destinație
  - c) procesul de actualizare a tabelului de rutare
  - d) procesul de configurare a unui protocol de rețea pe un ruter
3. Care dintre următoarele entități nu poate fi un câmp în tabela de rutare:
  - a) adresa destinație
  - b) masca de rețea
  - c) interfața următorului salt
  - d) protocolul de rutare
4. Un ruter dirijează pachetele IP pe baza:
  - a) adresei sursă și adresei destinație
  - b) adresei de rețea destinație și tabelului de rutare
  - c) adresei fizice
  - d) adresei destinație și protocolului DNS
5. Tabela de rutare a unui ruter static se configurează:
  - a) automat la fiecare pornire a acestuia
  - b) manual la instalarea ruterului în rețea
  - c) manual la fiecare pornire a sa
  - d) automat la fiecare modificare a topologiei de rețea
6. Ce comandă CISCO dată pe un ruter afișează fișierul de configurare memorat în NVRAM?
  - a) router# show flash
  - b) router# show nvram
  - c) router# show running-config
  - d) router# show startup-config
7. Ce port este folosit pentru configurarea inițială a unui ruter direct de la un calculator?
  - a) auxiliary
  - b) console
  - c) serial

d) fastEthernet

8. Ce comandă CISCO afișează tabela de rutare?

- a) ping
- b) show ip route
- c) show interface
- d) show table-config

9. Care este un deajavantaj al rutării dinamice?

- a) administrare greoaie
- b) necesită reconfigurare la fiecare schimbare de topologie de rețea sau de rutere
- c) nu poate ocoli defecțiunile apărute în rețea
- d) încarcă suplimentar traficul din rețea

10. Prin ce identifică un ruter rețeaua de destinație a unui pachet?

- a) prin adresa MAC
- b) prin adresa portului de destinație
- c) prin adresa IP destinație și masca de rețea
- d) prin adresa TCP

11. Care este rolul unei adrese de broadcast?

- a) de a transmite un pachet către o singură rețea destinație
- b) de a trimite un pachet către o listă de destinație
- c) de a trimite un mesaj către toate ruterele dintr-o rețea
- d) de a trimite un mesaj către toate gazdele dintr-o rețea/subrețea

12. Dacă o gazdă nu cunoaște o adresă de nivel 3 (o adresă IP) din cadrul acelei rețele, cum o poate afla?

- a) trimite un pachet ARP
- b) trimite un pachet RARP
- c) trimite un mesaj de avertizare administratorului rețelei
- d) trimite un mesaj de interogare spre serverul de nume (DNS)

13. Care dintre următoarele este o adresă multicast în IPv4?

- a) 111.111.111.111
- b) 255.255.255.255
- c) 224.0.0.251
- d) 224.0.0.2

14. Următoarea adresă IPv6 2023:0000:34FA:0000:0000:09C1:4322:AA43 se mai poate scrie corect astfel:

- a) 2023:0000:34FA::09C1:4322:AA43
- b) 2023:0000:34FA::9C1:4322:AA43
- c) 2023::34FA::9C1:4322:AA43
- d) 2023::34FA::09C1:4322:AA43

15. Interoperabilitatea IPv4 cu IPv6 se poate face astfel:

- a) Stiva duală
- b) Tunelare IPv6 peste IPv4

- c) Folosirea unor tabele de rutare duală
  - d) Protocoalele nu sunt interoperabile
16. Obiectivele urmărite prin trecerea la IPv6
- a) Să suplinească funcțiile protocoalelor de nivel transport
  - b) Să asigure mobilitatea gazdelor în rețea fără schimbarea adreselor
  - c) Să asigure servicii securizate la nivel rețea
  - d) Să asigure adrese de rețea pentru toți utilizatorii prezenți și viitori previzibili
17. Ce câmpuri obligatorii are antetul unui cadru Ethernet?
- a) Adresele IP sursă și destinație
  - b) Controlul erorilor (CRC)
  - c) Adresa MAC destinație
  - d) Numărul de secvență
18. Rolul unui server DNS într-o rețea de calculatoare este:
- a) Să stabilească nume de domenii și să le administreze
  - b) Să facă rezoluția *adresă fizică ↔ adresă IP*
  - c) Să întoarcă adresa IP a unei resurse din rețea desemnată prin numele din domeniu
  - d) Să atribuie adrese IP resurselor din rețea
19. Protocolul HTTP are rezervat portul:
- a) 80
  - b) 8080
  - c) 25
  - d) Nu are un port rezervat
20. Care dintre următoarele variante nu este un domeniu de prim nivel în Internet?
- a) ARP
  - b) COM
  - c) EDU
  - d) WINS
21. Ce comandă este mai des folosită pentru a afla informații despre configurarea IP în sistemele de operare Windows 7?
- a) PING
  - b) WINCONFIG
  - c) IPCONFIG
  - d) IP CONF
22. Pentru a vizualiza conexiunile active de pe un calculator conectat în rețea se poate folosi comanda *cmd*:
- a) SHOW CONFIG
  - b) NETSTAT
  - c) IPCONFIG/ALL
  - d) SH CONFIG
23. Ce aplicație din pachetul Windows 2008 Server administrează conturile utilizatorilor?
- a) Domain Controller-ul



- b) DHCP
- c) DNS
- d) IIS (Internet Information Server)

24. Cum este numit procesul prin care mai multe controlere de domeniu partajează copii complete ale bazei de date cu directoare de rețea și se actualizează constant între ele?

- a) Duplicare
- b) Replicare
- c) Încărcare
- d) Actualizare

25. Ce model de rețea este acela în care serviciile asigurate utilizatorilor la cererea acestora sunt găzduite de aplicații dedicate care rulează pe mașini fizice sau virtuale?

- a) Model bazat pe organizație
- b) Model bazat pe resurse
- c) Model client-server
- d) Model bazat pe calcul distribuit

26. Unde se configurează sau se șterge un adaptor de rețea sub sistemul de operare Windows XP?

- a) În miniaplicația Add/Remove Hardware
- b) În miniaplicația Network Adapter
- c) În miniaplicația Network
- d) Funcția Plug and Play rezolvă automat problema

27. Ce comandă se poate folosi pentru a afla numele serverului DNS dintr-o rețea?

- a) Winword
- b) NSLOOKUP
- c) PING
- d) NETSTAT

28. Câte adrese IP trebuie să aibă configurate un ruter pentru a face serviciul de rutare?

- a) Cel puțin una
- b) Una sau mai multe
- c) Două
- d) Cel puțin două în rețele diferite

29. Care dintre următoarele servere sunt servere web?

- a) Apache
- b) IIS (Internet Information Server)
- c) DNS
- d) Proxy Server

30. Care dintre protocoalele de mai jos permit administrarea resurselor hard inteligente în scopul cunoașterii stării lor funcționale?

- a) HTTP
- b) IMAP
- c) SNMP
- d) SMTP

31. Care dintre următoarele portocoale sunt folosite pentru crearea conexiunilor VPN?
- a) SIP (Session Initiation Protocol)
  - b) PPP (Point to Point Protocol)
  - c) SLIP (Serial Line Internet Protocol)
  - d) PPTP (Point to Point Tunneling Protocol)
32. Ce tehnici de detecție a erorilor se folosesc la nivelul legătură de date?
- a) Verificarea bitului de paritate
  - b) Verificarea ciclică a redundanței
  - c) Distanța Hamming
  - d) Retransmiterea cadrelor eronate
33. Standardul IEEE 802.11 se referă la:
- a) Rețele locale de calculatoare necablate
  - b) Rețele de arie largă de calculatoare necablate
  - c) Rețele de transport de date
  - d) Rețele de arie medie de calculatoare necablate
34. Care dintre metodele următoare tehnici evită coliziunile de date în LAN-uri?
- a) CSMA/CA
  - b) CSMA/CD
  - c) Token Ring
  - d) Token bus
35. O rețea din Internet are masca de subrețea 255.255.225.224. Care este numărul maxim de gazde din subrețea?
- a) 16
  - b) 30
  - c) 64
  - d) 32
36. Care dintre următoarele variante este o mască validă de subrețea fără clasă?
- a) 255.0.0.0
  - b) 255.255.255.0
  - c) 255.255.255.230
  - d) 255.255.255.240
37. Ce urmări are execuția comenzii Router#**erase nvram**
- a) Șterge istoria configurării ruterului
  - b) Șterge fișierul de configurare din NVRAM
  - c) Șterge configurarea curentă a ruterului
  - d) Comanda erase nu are efect asupra configurării
38. Comanda Router(config)#**hostname** permite:
- a) aflarea numelui ruterului
  - b) stabilirea numelui ruterului din modul configurare globală
  - c) accesul la configurarea interfețelor ruterului
  - d) stabilirea numelui și a parolei ruterului

39. Ce afirmații referitoare la tehnologia MPLS sunt adevărate:
- a) Este un mecanism de încapsulare suplimentar între nivelurile 2 și 3
  - b) Înlocuiește complet adresele IP dintr-o rețea
  - c) Folosește comutarea fluxurilor de date pe bază de etichete
  - d) Eticheta este un identificator de flux de lungime 20 biți
40. Tehnicile de transmisie cu spectru împrăștiat se folosesc cu preponderență:
- a) În rețelele Ethernet
  - b) În rețelele de transport
  - c) În rețelele FDDI
  - d) În rețelele wireless
41. Se dă adresa IP 140.20.37.133 și masca de subrețea 255.255.255.192. Să se precizeze adresa de subrețea din care face parte această adresă IP și adresa de difuzare în subrețea.
- a) 140.20.37.120 și 140.20.37.255
  - b) 140.20.37.128 și 140.20.37.191
  - c) 140.20.37.0 și 140.20.37.255
  - d) 140.20.37.128 și 140.20.37.120
42. Metrica unei rute poate fi:
- a) Un număr între 0 și 255 asociat unui protocol pentru ierarhizarea acestuia
  - b) Un criteriu sau un set de criterii de apreciere a calității acesteia
  - c) Numărul de salturi de la sursă la destinație
  - d) Sistem de priorități de transmitere a pachetelor de către un ruter
43. O rută direct conectată are distanța administrativă:
- a) 1
  - b) 250
  - c) 0
  - d) Distanța administrativă este un atribut al unui ruter, nu al unei rute
44. RIP este un protocol de rutare care:
- a) Folosește ca metrică întârzierea minimă în rețea
  - b) Se bazează pe starea legăturilor
  - c) Este folosit pentru rutare interdomeniu
  - d) Are numărul maxim de salturi egal cu 15
45. Care este matricea folosită de RIP pentru determinarea căii optime?
- a) Numărul minim de salturi până la destinație
  - b) Lățimea de bandă maximă până la destinație
  - c) Timpul minim până la destinație
  - d) Rata de eroare minimă până la destinație
46. Timpul de convergență al unui protocol reprezintă:
- a) Timpul dus-întors maxim dintre oricare pereche sursă-destinație din rețea
  - b) Timpul în care toate ruterele din rețea și-au configurat tabela de rutare
  - c) Timpul mediu de prelucrare a pachetelor de către un protocol de rețea
  - d) Timpul de viață al unui pachet în rețea
47. Serviciul de poștă electronică funcționează pe baza următoarelor protocoale mai cunoscute:

- a) SMTP, RTP, DNS
  - b) POP3, IMAP, SMTP**
  - c) IMAP, HTTP, SNMP
  - d) FTP, SNMP, SMTP
48. Citirea poștei electronice se poate face cu ajutorul următoarelor protocoale:
- a) SMTP, POP3
  - b) IMAP, SMNP
  - c) POP3, IMAP**
  - d) SMTP, SNMP
49. DHCP (Dynamic Host Configuration Protocol) este o aplicație client-server prin care:
- a) Serverul furnizează stației client numele de domeniu și adresa IP necesare funcționării în rețea
  - b) Se determină adresa IP corespunzătoare unui unei stații cu nume de domeniu cunoscut
  - c) Se atribuie adrese fizice stațiilor dintr-o rețea
  - d) Se face alocare dinamică de adrese IP stațiilor dintr-o rețea**
50. Comanda următoare: Router(config)#**router ospf 10** dată pe un ruter Cisco are ca efect:
- a) Validarea procesului OSPF pe acel ruter**
  - b) Definirea ID-ului de ruter necesar rulării OSPF
  - c) Definirea ariei 10 pe ruter
  - d) Validarea procesului OSPF și definirea ariei
51. Următoarea comandă de configurare a unui ruter  
Router(config-router)#**network 192.168.0.0 0.0 255.255 area 10** are ca effect:
- a) Activarea procesului OSPF pe acel ruter
  - b) Definește o interfață pe care rulează procesul OSPF și definește aria în care se află acea interfață**
  - c) Definește aria procesului OSPF
  - d) Defineste aria și numărul procesului OSPF
52. Care dintre următoarele protocoale suportă rutarea ierarhică?
- a) RIPv1 și OSPF
  - b) RIPv3 și OSPF**
  - c) OSPF și RIPv2**
  - d) RIPv1 și IGRP
53. Interoperabilitatea IPv4 cu IPv6 pe baza stivei duale presupune:
- a) Ambele stive de protocoale sunt activate pe interfețele unui ruter**
  - b) Configurarea unui tunel IP între echipamentele terminale ale aplicației
  - c) Se activează automat unul sau altul dintre protocoale în funcție de formatul cadrelor recepționate de la nivelul legătură de date
  - d) Stiva duală nu este o posibilitate de interoperabilitate IPv4 – IPv6
54. Următoarea adresă IPv6 FE80::200:CFF:FE3A:8B18 este:
- a) Adresă multicast
  - b) Adresă link local**
  - c) Adresă site local

d) Adresa broadcast

55. O adresa IPv6 de tip link local:

- a) Este configurată automat pe interfața unei gazde îndată ce protocolul IPv6 a fost validat
- b) are formatul de prefix fc00::/7
- c) Este obligatorie pentru comunicarea pe un link local între două dispozitive IPv6
- d) Este vizibilă și folosită pentru rutare de oriunde din Internet

56. Adresele IPv6 unicast agregabile globale:

- a) Au formatul de prefix 2000::/3
- b) Sunt structurate ierarhic
- c) Au lungimea totală de 64 biți
- d) Nu sunt rutabile în Internet

57. Protocolul ICMP este folosit de rețelele IPv4 pentru:

- a) Rutare în Internet
- b) Descoperirea gazdelor conectate în rețea
- c) Configurarea tabelelor de rutare
- d) Raporta problemelor legate de livrarea pachetelor în rețea

58. Protocolul BGP este folosit pentru rutarea datelor:

- a) În rețelele de bandă largă
- b) În rețele locale de calculatoare
- c) Între sisteme autonome
- d) În rețelele de organizație

59. Care dintre afirmațiile următoare despre un agent SNMP sunt adevărate :

- a) Solicită informații de management de la stația de management
- b) Răspunde interogărilor de management primite de la sistemul de management
- c) Este un protocol de nivel rețea
- d) Ține o bază de date cu informațiile de management din echipamentul în care este resident

60. O comunitate SNMP este:

- a) Totalitatea elementelor de rețea administrate prin SNMP
- b) Un mecanism de securitate pentru comunicația dintre sistemul de management și agentul de management
- c) O rețea în care este instalată aplicația SMNP
- d) Totalitatea obiectelor administrate prin SNMP

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Criptografie si securitatea informatiei**

# CRİPTOGRAFIE

1. Care este rolul coderului de sursă?
  - a. Criptarea textului generat de sursa informațională;
  - b. Compresia informației generate de sursa informațională;
  - c. Schimbarea formei de reprezentare a informației generate de sursă;
  - d. Protejarea informației generate de sursă împotriva perturbațiilor.
2. Care este rolul coderului de canal?
  - a. Compresia informației generate de sursa informațională;
  - b. Schimbarea formei de reprezentare a informației generate de sursă;
  - c. Protejarea informației împotriva perturbațiilor din canalul de transmitere a acesteia;
  - d. Criptarea textului generat de sursa informațională.
3. Care este rolul blocului de criptare informațională?
  - a. Schimbarea formei de reprezentare a informației generate de sursă;
  - b. Criptarea textului generat de sursa informațională;
  - c. Prelucrarea informației în scopul păstrării confidențialității transmisiei;
  - d. Compresia informației generate de sursa informațională.
4. Cheia criptografică este :
  - a. Subansamblu hardware utilizat pentru protecția software;
  - b. Expresie matematică care reprezintă modul de funcționare al blocului de criptare și de decriptare;
  - c. Parametrizare a funcției criptografice implementate în blocurile de criptare/decriptare;
  - d. Șir parametric de valori statice sau dinamice care particularizează comportamentul funcției criptografice.
5. Un sistem criptografic pentru transmiterea/recepția informației se compune din:
  - a. Bloc de criptare și bloc de decriptare cu chei criptografice diferite;
  - b. Bloc de criptare și bloc de decriptare fără chei criptografice;
  - c. Bloc de criptare și bloc de decriptare cu chei criptografice identice;
  - d. Bloc de criptare cu două chei criptografice diferite.
6. Decriptarea reprezintă:
  - a. Operația efectuată de către un intrus pentru a descoperi mesajul clar dintr-un text criptat;
  - b. Operația efectuată de către destinatar pentru a extrage mesajul clar dintr-un text criptat recepționat;

## CRİPTOGRAFIE

- c. Operația efectuată de către un intrus și/sau destinatar pentru a descoperi mesajul clar dintr-un text criptat;
- 7. Domeniul de știință al criptologiei are următoarele laturi:
  - a. **Criptografia;**
  - b. Steganografia;
  - c. **Criptanaliza;**
  - d. Steganaliza.
- 8. Criptanaliza reprezintă:
  - a. Prelucrarea informației efectuată de către destinatar pentru a extrage mesajul clar dintr-un text criptat recepționat;
  - b. **Ansamblul de prelucrări matematice și statistice efectuate de către un interceptor neautorizat pentru a descoperi mesajul clar dintr-un text criptat;**
  - c. **Interpretări bazate pe analize empirice și/sau științifice, efectuate de către un interceptor neautorizat, pentru a descoperi mesajul clar dintr-un text criptat.**
- 9. Tehnici criptografice recomandate de Claude Shannon pentru optimizarea algoritmilor moderni:
  - a. Tehnica compresiei;
  - b. **Tehnica confuziei;**
  - c. **Tehnica difuziei;**
  - d. Tehnica propagării erorii criptografice.
- 10. În funcție de complexitatea mașinilor criptografice acestea pot fi clasificate în:
  - a. Monoalfabetice;
  - b. **Polialfabetice simple;**
  - c. **Polialfabetice cu autocifrare;**
  - d. **Polialfabetice complexe.**
- 11. Cel mai important parametru caracteristic pentru complexitatea mașinilor criptografice este:
  - a. Viteza de transfer a informației;
  - b. **Timpul de acoperire;**
  - c. Numărul de alfabete utilizate;
  - d. Mărimea cheii criptografice.
- 12. Dimensiunea cheii criptografice determina:
  - a. **Mulțimea cheilor utilizabile posibile;**
  - b. **Viteza de decriptare;**
  - c. **Complexitatea sistemului criptografic;**



## CRİPTOGRAFIE

- d. Timpul de acoperire al sistemului criptografic.
13. Reușita unui atac criptanalitic este influențată de:
- a. **Experiența criptanalistului;**
  - b. Lungimea canalului de transmitere a informației;
  - c. **Mijloacele tehnice de care dispune;**
  - d. **Volumul de text cifrat aflat la dispoziția criptanalistului.**
14. Cel mai vechi dispozitiv criptografic, datând din timpul războaielor spartane a fost:
- a. Trifidul lui Delastelle;
  - b. **Skytala;**
  - c. Alfabetul lui Collon;
  - d. Cilindrul Bazaries.
15. Mașina Enigma este o mașină criptografică:
- a. Polialfabetică simplă;
  - b. Polialfabetică cu autocifrare;
  - c. **Polialfabetică complexă;**
  - d. Polialfabetică cu autocheie.
16. Metoda de cifrare Vernam se bazează pe:
- a. Funcții neliniare;
  - b. Aplicarea principiilor de difuzie și confuzie ;
  - c. **Sumarea modulo 2, bit cu bit, a textului în clar cu cheia de cifrare;**
  - d. Operații polialfabetice complexe.
17. Un adversar a obținut, în urma unei interceptări radio, următorul text cifrat “WULPLWH DUPDWD”. Pe baza unui atac de criptanaliză prin forță brută (Cifrul Caesar – alfabet 26 de caractere), adversarul va obține următorul text clar:
- a. ATACATI IN FLANC;
  - b. FOLOSITI REZERVA;
  - c. **TRIMITE ARMATA;**
  - d. RETRAGE ARMATA.
18. Presupunem că Alice dorește să transmită lui Bob mesajul “TRIMIT AJUTOR”, folosind un canal de comunicație nesigur. Atât ea cât și Bob au obținut printr-o metodă necunoscută o cheie de cifrare generată aleator și anume: “AFVKGHOPERTS”. Pentru a cifra mesajul, Alice va utiliza cifrul VERNAM cu operații mod 26. În urma operației de cifrare mesajul primit de Bob va fi:
- a. KFDDSJSKLCBNDA;
  - b. AWEDKFODOESOE;
  - c. **TWDWAOYOKHJ;**

## CRİPTOGRAFIE

- d. JCVSKFEOJNFEEFK.
19. Sistemul Jammet realizat de Jammet Onde Electrique în anul 1926, este:
- a. Analog sistemului Vernam bazându-se însă pe alfabetul MORSE;
  - b. Derivat din mașina de cifrat ENIGMA;
  - c. Inspirat din sistemul Belin de cifrare a imaginilor;
  - d. Este un sistem de codare.
20. Managementul cheilor de cifrare este un proces complex care constă în:
- a. Generarea, distribuția, utilizarea și distrugerea cheilor de cifrare;
  - b. Folosirea canalului nesigur pentru transmiterea cheilor de cifrare;
  - c. Transmiterea pe INTERNET a cheilor de cifrare;
  - d. Instalarea echipamentelor de cifrare.
21. Distribuția manuală a cheilor de cifrare presupune:
- a. Transmiterea electronică în clar a acestora ;
  - b. Distribuția cheilor, sub formă tipărită, pe canale publice (dacă sunt cifrate) sau pe canale private (cu grad maxim de siguranță);
  - c. Folosirea serviciilor de e-mail publice;
  - d. Compresarea acestora.
22. Într-un sistem de cifrare se poate utiliza, în principiu, o combinație complexă de chei de cifrare, astfel:
- a. Cheie de cifrare elementară (de structură), cheie secundară (care particularizează algoritmul de cifrare) și cheie de lucru, toate obținute cu ajutorul unui generator aleator de zgomot;
  - b. Nu se folosesc chei de cifrare;
  - c. Cheile de cifrare sunt generate prin însumarea unor factori constanți cunoscuți;
  - d. Cheile de cifrare sunt generate de algoritmul de cifrare.
23. Definiția dată de Shannon pentru un sistem de cifrare este:
- a. Un sistem de cifrare realizează o compresie a sursei de date;
  - b. Un sistem de cifrare folosește chei de cifrare pentru compresia sursei de date;
  - c. Un sistem de cifrare este o familie  $T$  de transformări inversabile, dintr-o mulțime de mesaje  $M$  într-o mulțime de criptograme  $C$ ;
  - d. Un sistem de cifrare folosește algoritmul Shannon-Fano pentru realizarea operațiunii de cifrare.

## CRIPTOGRAFIE

24. În scopul evaluării securității unui sistem de cifrare, precizați care din următoarele enunțuri este o condiție fundamentală:
- a. Criptanalistul are cunoștințe complete despre sistemul de cifrare, a acumulat un volum considerabil de text cifrat și are la dispoziție un anumit volum de text clar și echivalentul său cifrat;
  - b. Criptanalistul cunoaște o parte din cheile de cifrare ale sistemului;
  - c. Criptanalistul cunoaște modul de funcționare a mașinii de cifrat analizate;
  - d. Criptanalistul cunoaște sistemul de cifrare și cheile utilizate.
25. Un sistem secret cu acoperire unică (cifru cu cheie unică sau cifru Vernam) se bazează:
- a. pe tehnica „confuziei”;
  - b. pe tehnica „difuziei”;
  - c. pe tehnica „confuziei” și pe tehnica „difuziei”;
  - d. pe utilizarea de funcții neliniare.
26. Care din următoarele enunțuri nu este principiu care să stea la baza construcției unui sistem de cifrare:
- a. Mărimea cheii de cifrare;
  - b. Simplitatea operațiunilor de cifrare și descifrare;
  - c. Minimizarea cheii de cifrare;
  - d. Cantitatea de secretizare oferită.
27. Un sistem de cifrare este *strict ideal* dacă:
- a.  $H(X/C) > H(K)$ ;
  - b.  $H(X/C) < H(K)$ ;
  - c.  $H(X/C) \leq H(K)$ ;
  - d.  $H(X/C) = H(K)$ .
28. *Testul spectral* se bazează pe:
- a. testele de bit seriale;
  - b. probabilitățile de apariție a unor secvențe constante;
  - c. tehnica transformărilor Fourier aplicate funcțiilor de variabile întregi, transformări care permit scoaterea în evidență a caracterului aleator al unei secvențe de numere întregi;
  - d. Analiza factorilor constanți din succesiunile pseudoaleatoare.
29. Dacă notăm cu  $n_0$  numărul de biți de „0” și cu  $n_1$  numărul de biți de „1” dintr-o succesiune de biți, atunci, conform testului de frecvență, pentru a avea o succesiune aleatoare, este necesar ca:
- a.  $n_0 \cong n_1$ ;
  - b.  $n_0 < n_1$ ;
  - c.  $n_0 > n_1$ ;
  - d.  $n_0 \ll n_1$ ;

## CRIPTOGRAFIE

30. În teoria criptanalizei, prin *spargerea* unui sistem de cifrare  $S$  se înțelege:
- evidențierea textelor clare pe baza analizelor criptogramelor realizate prin intermediul lui  $S$ , fără a avea cunoștințe despre cheia folosită, dar cunoscând sistemul utilizat;
  - evidențierea textelor clare pe baza analizelor criptogramelor realizate prin intermediul lui  $S$ , fără a avea cunoștințe despre sistemul utilizat și cheia folosită;
  - evidențierea textelor clare pe baza analizelor criptogramelor realizate prin intermediul lui  $S$ , fără a avea cunoștințe despre sistemul utilizat, dar cunoscând cheia folosită;
  - evidențierea textelor clare pe baza analizelor criptogramelor realizate prin intermediul lui  $S$ , având cunoștințe despre sistemul utilizat și cheia folosită.
31. Într-un sistem secret, informația este complet ascunsă atunci când:
- $H(C) > \log n$
  - $H(C) < \log n$
  - $H(C) = \log n$ .
32. Prin tehnica “confuziei”, utilizată la proiectarea algoritmilor de cifrare, se urmărește:
- Corelarea cât mai complexă între criptogramă și cheie;
  - Corelarea cât mai complexă între mesajul în clar și cheie;
  - Corelarea cât mai complexă între mesajul în clar și criptogramă.
33. Care dintre afirmațiile de mai jos nu reprezintă un criteriu de aleatorism propus de Golomb:
- Dacă  $p$  este par atunci secvența de lungime  $p$  va avea  $p/2$  biți de valoare 0 și  $p/2$  biți de valoare 1. Dacă  $p$  este impar, atunci numărul de *zerouri* și numărul de cifre *unu* diferă printr-o unitate;
  - Într-o secvență de lungime  $p$ , jumătate din numărul seriilor va avea lungimea 1,  $1/4$  va avea lungimea 2,...,  $1/2^k$  din numărul seriilor va avea lungimea  $k$ ;
  - Autocorelația defazată nu este constantă.
34. Pentru evaluarea caracteristicilor aleatoare a unui șir de numere, *testul spectral* se bazează pe:
- Tehnica transformărilor Fourier;
  - Tehnica evaluării numărului de apariții a biților de 0 și de 1;
  - Tehnica transformărilor complexe.
35. În criptanaliză, cuvintele probabile sunt:
- Cuvinte care pot fi întâlnite într-un mesaj, ca urmare a faptului că sunt caracteristice pentru sursa respectivă;
  - Litere și cifre care apar în mod pseudoaleator;
  - Mesaje informaționale care au probabilitate de apariție de 0.5.

## CRIPTOGRAFIE

36. Testul SERIAL este utilizat pentru:
- a. Analiza probabilităților de tranziție dintre două cifre binare succesive din șir, care pot fi aceleași sau foarte apropiate;
  - b. Verificarea dependenței fiecărui simbol de predecesorul său;
  - c. Seriile de câte doua simboluri din cadrul șirului sunt independente de predecesoarele acestora.
37. In cadrul sistemului Vernam codificarea cheilor si ale textului clar se realizează cu codul:
- a. Baudot;
  - b. Morse;
  - c. Baudot și Morse.
38. Metodele de generare ale cheilor criptografice sunt:
- a. Metode algebrice;
  - b. Metode aritmetice;
  - c. Metode bazate pe teoria haosului;
  - d. Metode empirice;
  - e. Metode bazate pe fenomene fizice.
39. Care dintre următoarele sisteme criptografice sunt sisteme tomogramice:
- a. Jammet
  - b. Belin
  - c. Bazaries
  - d. Vernam
40. Sistemele hardware de generare a șirurilor pseudoaleatoare pot fi bazate pe:
- a. Scheme secvențiale liniare;
  - b. Registre de deplasare;
  - c. Generatoare haotice;
  - d. Scheme secvențiale neliniare.
41. Cifrurile tomogramice au la bază tehnici care urmăresc:
- a. Fraționarea cuvintelor din textul clar;
  - b. Fraționarea literelor din textul clar;
  - c. Fraționarea literelor din textul criptat;
  - d. Fraționarea textului clar în propoziții simple cu sens.
42. Care din următoarele tehnici de prelucrare informațională nu sunt cifruri tomogramice:
- a. Alfabetul Baudot;
  - b. Sistemul Vernam;
  - c. Cifrul lui Cezar;
  - d. Codul Morse.

## CRİPTOGRAFIE

43. Semnătura digitală are la bază următoarele tehnici:
- a. **Criptarea asimetrică;**
  - b. Criptarea simetrică;
  - c. Digitizarea semnăturii olografe;
  - d. Reproducerea semnăturii olografe pe documentele electronice.
44. Care dintre următoarele enunțuri fac parte din setul de condiții (principii) enunțate de Claude Shannon cu privire la realizarea sistemelor criptografice:
- a. **Cantitatea de secret conferită mesajului criptat;**
  - b. Nerepudierea mesajelor;
  - c. **Optimizarea mărimii cheilor criptografice;**
  - d. **Evitarea propagării erorilor în textul criptat.**
45. Metodele de criptare aplicate mesajelor pot fi:
- a. **Metode criptografice simetrice;**
  - b. **Metode criptografice asimetrice;**
  - c. Metode criptografice bazate pe chei aleatoare;
  - d. **Metode criptografice bazate pe funcții de permutare .**
46. Un sistem informațional binar echiprobabil are entropia egală cu :
- a. **1 bit / eveniment;**
  - b. 0 bit / eveniment;
  - c. 2 bit / eveniment;
  - d. 1/2 bit / eveniment.
47. Entropia unui sistem informațional poate fi:
- a. **Nulă;**
  - b. **Pozitivă;**
  - c. Strict pozitivă;
  - d. Negativă.
48. Timpul de acoperire al unui sistem criptografic este:
- a. Timpul de valabilitate al sistemului de chei;
  - b. **Timpul în care un sistem criptografic poate fi considerat sigur;**
  - c. Timpul necesar pentru a fi decriptat un mesaj;
  - d. Timpul în care se realizează criptarea mesajelor.
49. Tehnica codificării redundante este:
- a. Tehnică criptografică;
  - b. Tehnică de compresie a surselor informaționale;

## CRIPTOGRAFIE

- c. Tehnică de protecție la erori a canalelor cu perturbații;
  - d. Tehnică de protecție la erori a canalelor fără perturbații.
50. Codificarea neuniformă este utilizată ca:
- a. Tehnică criptografică;
  - b. Tehnică de compresie a surselor informaționale;
  - c. Tehnică de protecție la erori a canalelor cu perturbații;
  - d. Tehnică de protecție la erori a canalelor fără perturbații.

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**MODULUL 4**

***Tehnologii web și comerț electronic***



**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Tehnologii web**

## TEHNOLOGII WEB

### 1. Care este rolul tagului `<keygen>` din HTML 5?

- a. marchează text evidențiat;
- b. marchează un grup de elemente care au legatura unul cu celălalt;
- c. **marchează un cod generat automat într-un formular;**
- d. marchează diferite tipuri de rezultate ale unui script oarecare ;

### 2. Care sunt attribute pentru tagul `input` din HTML 5?

- a. **accept;**
- b. **autocomplete;**
- c. `day_time` ;
- d. `form_action`;

### 3. Atributul `name` este utilizat pentru tagurile?

- a. `linked`, `textarea`, `select`;
- b. `form_name`, `table`, `a`, `img`;
- c. `input`, `area`, `a`, `iframe`, `name`;
- d. **`form`, `input`, `frame`, `a` ;**

### 4. Care din următoarele taguri sunt definite corect?

- a. `<link rel="stylesheet" type="text/css" href="exemplu.css" />` ;
- b. `<?xml-stylesheet href="exemplu.css" type="text/css"?>`;
- c. `<area shape="polygon" coords="50,350,150,350,100,250,300" href="test.html">` ;
- d. **`<area shape="circle" coords="100,100,30" href="cv.html" target="_blank">`;**

### 5. Primul fișier imagine încărcat într-o pagină html poate fi schimbat cu un alt fișier dacă se utilizează:

- a. `window.images[1].src="poza.jpg"`;
- b. `document.status.images[1].src="poza.jpg"`;
- c. `document.images[1].src="poza.jpg"`;
- d. **`document.images[0].src="poza.jpg"`;**

### 6. Imaginea de fundal a unei pagini Web se stabilește astfel?

- a. `<img background="p.gif">`;
- b. **`<body background="image/imagine.gif">`;**
- c. `<body bgcolor="p.gif">`;
- d. `<body background-image="image/imagine.gif">`;

### 7. Dimensionarea unei imagini se realizează folosind următoarele attribute ale tag-ului `<img>`?

- a. `width` și `align`;
- b. `align` și `height`;
- c. **`width` și `height`;**
- d. `width`, `height` și `size`;

### 8. Care din următoarele afirmații sunt adevărate?

- a. **Atributul `href` se poate asocia etichetei `a`;**
- b. Atributul `padding_cell` aparține etichetei `<table>` și setează distanța dintre celulele tabelului;
- c. **`&nbsp;` - introduce un spațiu;**
- d. **Atributul ACTION al marcatului FORM stabilește adresa fișierului la care sunt trimise datele dintr-un formular;**

**9. Transmiterea datelor dintr-un formular către un fișier de pe server, se realizează?**

- a. `<input type="reset" value="trimite">;`
- b. `<input type="button" value="trimite"> ;`
- c. `<input type="submit" value="apasa"> ;`
- d. `<input name="submit" value="transmite">;`

**10. Pentru a trimite valoarea „html” cu ajutorul unei casete de validare checkbox se folosește:**

- a. `<input type="checkbox" name="c" />html ;`
- b. `<input type="checkbox" value="html" name="c" checked />trimite ;`
- c. `<input value="html"/>html ;`
- d. `<input type="checkbox" value="html" name="value_html" checked />html;`

**11. O clasă de stiluri se apelează prin?**

- a. `<DIV ID="nume_clasa">`
- b. `<DIV CLASS="nume_clasa">`
- c. `<DIV STYLE="nume-clasa">`
- d. `clasa_stil {.....}`

**12. Cum se definește un stil care afișează textul de mărime 10pt și culoare albă?**

- a. `.stil {font-size:10pt; color: white;}`
- b. `#stil {text-size:10px; color: green;}`
- c. `#stil {font-size:10pt; color: #FFFFFF;}`
- d. `.stil {font-size:10px; color: #000000;}`

**13. Care afirmații sunt adevărate?**

- a. *Un script JavaScript poate fi introdus în interiorul marcajelor `<script></script>`*
- b. *Funcțiile JavaScript definite de utilizator pot fi și într-un fișier cu extensia .js*
- c. *O clasă de stiluri este definită corect astfel:*  
`#clasa {color:#ab1121;background-image:url("poza.gif");text-decoration:underline;}`
- d. `<link rel="stylesheet" type="text/css" href="exemplu.css" />`

**14. Care este rezultatul următoarei secvențe din JavaScript "1" + 2 + 4 ?**

- a. 124
- b. "124"
- c. "1" și 24
- d. "1" și 6

**15. În Javascript obiectul Date se creează?**

- a. `data = new Array();`
- b. `data = new Object();`
- c. `data = new Date();`
- d. `Date= new data();`

**16. În Javascript numărul zilei din lună se extrage apelând metoda?**

- a. `getDate()`
- b. `getDay()`
- c. `getMonth()`
- d. `setDay()`

**17. Metodele și proprietățile obiectului String în JavaString sunt:**

- a. *substr()*
- b. *replace()*
- c. *length\_string()*
- d. *charCodeAtCh()*

**18. Alegeți evenimente din Javascript?**

- a. *OnKeyDown*
- b. *OnErrorFocus*
- c. *onLoad*
- d. *onUnload*

**19. Care sunt metodele și proprietățile pentru obiectul window din Javascript?**

- a. *open()*
- b. *status*
- c. *alert()*
- d. *write()*

**20. Tipurile de date utilizate în PHP sunt?**

- a. *String*
- b. *Array*
- c. *Null*
- d. *Data*

**21. Care din următoarele afirmații sunt adevărate?**

- a. *\$GLOBALS* cuprinde referințe spre orice variabilă globală care este accesibilă scriptului PHP curent;
- b. *\$\_SERVER* conține o serie de variabile ale căror valori sunt setate de server-ul web.
- c. *\$\_COOKIE* conține valorile variabilelor care cuprind informații referitoare la cookie-urile păstrate pe calculatorul utilizatorului care accesează pagina web.
- d. *\$\_SESSION* cuprinde variabile disponibile prin intermediul oricărui tip de mecanism cu ajutorul căruia utilizatorul poate introduce date.

**22. Ca va afișa scriptul PHP:**

```
<?php
$i = 5.7;
echo $i."<br />";
$i = sprintf("%05.2f", $i);
echo $i;
?>
```

- a) 5.7;05.70
- b) 5.7  
05.70
- c) 5.70  
05.7
- d) 5.7 05.7

**23. Care funcții PHP acționează asupra fișierelor?**

- a. *fgetc()* ;
- b. *is\_file()* ;
- c. *filesize()*;
- d. *close\_is()*;

**24. Menționați metodele care nu fac parte din obiectul `String` pentru JavaScript?**

- a. `toLowerCase()`;
- b. `setTimeout()`;
- c. `focus()`;
- d. `strike()`;

**25. Un document XML:**

- a. este o colecție de entități;
- b. trebuie să fie construit corect;
- c. nu este colecție de date;
- d. nu este sensibil la tipul literelor;

**26. DTD-ul (Document Type Definition) conține:**

- a. numele efectiv al elementelor;
- b. regulile utilizate pentru a determina care elemente se pot imbrica în alte elemente;
- c. structura documentului împreună cu lista elementelor permise.
- d. codificările de limbă utilizate în document;

**27. O schema XML se generează dintr-o schemă relațională astfel:**

- a. pentru fiecare tabelă se generează un tip de element;
- b. pentru fiecare coloană se adaugă la model un atribut la tipul elementului sau un element descendent ce conține numai PCDATA;
- c. pentru fiecare element se atasează o bază de date;
- d. Nu se pot defini scheme XML decât dacă se utilizează și PHP-ul;

**28. Ce rol are argumentul `secure` (sigur) pentru funcția `setcookie()` din PHP?**

- a. indică dacă o variabilă cookie trebuie trimisă prin intermediul unei conexiuni HTTP;
- b. indică dacă o variabilă cookie trebuie trimisă prin intermediul unei conexiuni sigure (HTTPS);
- c. permite identificarea numelui de domeniu asociat, dacă conexiunea nu este sigură;
- d. variabila cookie este disponibilă indiferent de tipul de conexiune utilizată;

**29. Cum se definește o constantă în PHP?**

- a. prin directiva `define()`;
- b. cu ajutorul cuvântului cheie `const`;
- c. cu ajutorul cuvintelor cheie `var const`;
- d. cu ajutorul cuvântului `static`;

**30. Cum poate fi accesat conținutul unei casete de text dintr-un formular, folosind JavaScript?**

- a. `nume_formular.nume_camp.value`
- b. `nume_camp.nume_formular.value()`
- c. `value().nume_formular.nume_camp`
- d. `value.nume_camp.nume_forumar`

**31. Care este sintaxa corectă Javascript pentru a scrie "Hello World"?**

- a. `document.write("Hello World ");`
- b. `alert.write("Hello World");`
- c. `window.write("hello world");`
- d. `document.write(a=" Hello World ");`

**32. Cum se scrie "Hello World" într-o fereastră alert?**

- a. alertBox="Hello World";
- b. alert=("Hello World");
- c. msgBox("Hello World");
- d. **alert("Hello World");**

**33. Cum se scrie "Hello World" în PHP?**

- a. Document.Write("Hello World");
- b. document.write("Hello World");
- c. **echo "Hello World";**
- d. **\$mes="Hello World";echo \$mes;**

**34. Variabilele în PHP încep cu:**

- a. **\$**
- b. **&**
- c. **!**
- d. **?php**

**35. Care este sintaxa unei reguli CSS?**

- a. **selector {proprietate : valoare;}**
- b. selector(proprietate, valoare)
- c. selector.proprietate = valoare;
- d. selector(proprietate) = valoare;

**36. Ce înseamnă PHP?**

- a. Private Hypertext Processor;
- b. **PHP: Hypertext Preprocessor;**
- c. Private Home Page;
- d. Personal Home Page;

**37. Care este modul corect pentru a ne conecta la o bază de date MySQL prin PHP?**

- a. mysql\_open("localhost","user","parola");
- b. connect\_mysql("localhost","user","parola");
- c. **mysql\_connect("localhost","user","parola");**
- d. dbconnetc("localhost");

**38. Care este modul corect de a adăuga 1 la variabila \$numara?**

- a. \$numara =+1;
- b. ++numara;
- c. **\$numara++;**
- d. numara++;

**39. Alegeți varianta corectă pentru definirea unei funcții Javascript:**

- a. function suma nr() {...}
- b. **function suma\_nr(){...}**
- c. suma\_nr();
- d. function suma\_nr { ... };

**40. Care afirmații sunt adevărate?**

- a. XML-ul nu este un limbaj de programare;
- b. Sunt compilatoare XML care să citească/scrie/actualizeze un document XML
- c. XML nu este un protocol pentru transmiterea datelor în rețea;
- d. Un document XML poate fi citit de orice parser.

**41. Pentru a introduce un fișier RSS în paginile unui site trebuie specificat în partea de HEAD a documentului web o linie de formă:**

- a. `<link rel="alternate" type="application/rss+xml" title="RSS" href="http://cale.. " />`
- b. `<link rel="" type="text/rss" href="exemplu.css" />`
- c. `<?xml-stylesheet href="exemplu.css" type="text/css"? />`
- d. `<link rel="stylesheet/alternate" type="application/rss+xml" title="RSS" href="http://cale.. " .>`

**42. Care este rolul butonului "submit" pentru codul de mai jos?**

`<form action="test.php" method="post">`

Introduceți numele : `<input type="text" name="n" /><br />`

`<input type="submit" value="Trimite" />`

`</form>`

- a. Transmite la server variabila n din formular cu metoda implicită;
- b. Transmite la fișierul test.php de pe server conținutul variabilei n;
- c. Transmite cu metoda post, la fișierul test.php de pe server conținutul variabilei n și readuce formularul la starea inițială;
- d. Transmite cu metoda implicită post, la fișierul test.php de pe server, conținutul variabilei n și readuce formularul la starea inițială;

**43. Proprietatea checked atașat unui element radio dintr-un formular are ca efect?**

- a. Bifarea implicită a respectivului element radio;
- b. Nu există acest atribut;
- c. Trebuie atașat obligatoriu la toate elementele radio din același grup;
- d. Se folosește doar pentru elemente de tip checkbox;

**44. Un fișier "script.js" este folosit într-un document html astfel:**

- a. `<script type="text/javascript" src="script.js"></script>`
- b. `<script href="text/javascript" src="script.js"></script>`
- c. `<script type="text/javascript" link="script.js"></script>`
- d. `<script rel="text/javascript" link="script.js"></script>`

**45. Care sunt atributele etichetei <textarea>?**

- a. cols - specifică numărul de caractere afișate într-o linie;
- b. rows - specifică numărul de linii afișate simultan;
- c. name - permite atașarea unui nume câmpului de editare multilinie;
- d. text-wrap - determină comportamentul câmpului de editare față de sfârșitul de linie.

**46. Care afirmații sunt adevărate cu privire la tratarea evenimentelor în JavaScript?**

- a. OnFocusError - acțiune care permite oprirea încărcării unei imagini
- b. OnFocus - acțiune ce va fi declansată de focalizarea asupra unui element.
- c. OnLoad - acțiune ce va fi declansată la încărcarea unui document.
- d. OnBlurLoad - acțiune prin care elementul primește focus

**47. Ce rol are funcția explode(\$separator, \$sir) în PHP?**

- preia valorile din variabila \$sir în care acestea sunt delimitate prin variabila \$separator și le plasează într-un tablou/vector;
- preia valorile dintr-un vector și le reunește într-un șir;
- afiseaza de cate ori apare variabila \$separator în variabila \$sir;
- Concateneaza variabilele \$separator și \$sir;

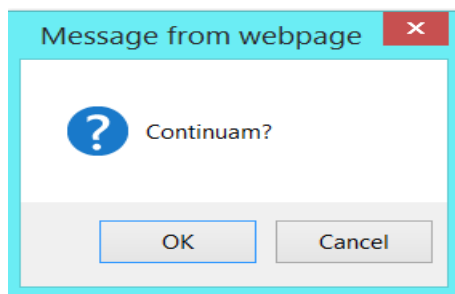
**48. Care afirmatii sunt adevarate pentru folosirea imaginii SVG(Scalable Vector Graphics) în HTML 5:**

- Imaginile nu pot fi create cu editoare text;
- Imaginile pot fi indexate în motoarele de căutare;
- Imaginile pot fi scalabile sau comprimate;
- Imaginile pot fi printate la orice rezoluție fără să se piardă din calitate;

**49. Ce rol au tagurile enumerate mai jos, în HTML5?**

- <audio> - specifică un fișier audio;
- <command> - specifică o comandă pe care un utilizator o poate invoca;
- <datalistno> - specifică o listă creată dinamic;
- <embed> - specifică conținut extern;

**50. Pentru a obține în Internet Explorer fereastra de mai jos, care este secventa de cod utilizata?**



- `<script type="text/javascript">  
test = window.confirm("Continuam?");  
if (test) { window.alert("Corect"); } else { window.alert("Incorect"); }  
</script>`
- `<script type="text/javascript">  
test = window.alert("Continuam?");  
if (test) window.confirm("Corect"); else window.alert("Incorect");  
</script>`
- `<script>  
test = window.alert("Continuam?");  
if (test) window.confirm("Corect"); else window.confirm("Incorect");  
</script>`
- `<script>  
test = confirm("Alege");  
if (test) window.confirm("Continuam"); else window.confirm("Incorect");  
</script>`

**51. Ca va afisa scriptul PHP ?**

```
<?php
$a = array(1=>"elem 1", 2=>"elem 1",3=>"elem 1"); foreach ($a as $index => $mesaj)
{   echo "<br />$index =>$mesaj"; }
?>
```



a.

1=elem 1

2=elem 2

3=elem 3

b.

1 =>elem 1

2 =>elem 1

3 =>elem 1

c.

1=elem 1

2=elem 1

3=elem 1

d.

1 =>elem 1

1 =>elem 1

1 =>elem 1

## 52. Ca va afisa scriptul PHP ?

```
<?php
$sir = "WEB, GRAFICA, PROGRAMARE C++, JAVA";
$matrice = explode(" ", $sir);
foreach($matrice as $key=>$nume) { echo $nume."<br />"; }
?>
```

a.

WEB

GRAFICA

PROGRAMARE C++

JAVA

b.

WEB GRAFICA PROGRAMARE C++ JAVA

c.

WEB, GRAFICA, PROGRAMARE C++, JAVA

d.

WEB,

GRAFICA,

PROGRAMARE C++,

JAVA

## 53. Ca va afisa scriptul PHP ?

```
<?php
$matrice = array("WEB", "GRAFICA", "PROGRAMARE C++", "JAVA");
sort($matrice);
$sir = implode(" ", $matrice);
echo "Cursuri: ".$sir;
?>
```

a. Cursuri: "WEB", "GRAFICA", "PROGRAMARE C++", "JAVA

b. Cursuri: WEB, GRAFICA, PROGRAMARE C++, JAVA

c.

Cursuri: WEB

GRAFICA

PROGRAMARE C++

JAVA

**54. Ca va afisa scriptul PHP ?**

```
<?php
function arie_dr($lungime, $latime) {
    return $lungime * $latime;
}
$rezultat = arie_dr(2,3);
echo "L*1 =" . $rezultat;
?>
```

- a) L\*1=6
- b) =6
- c) adrie\_dr(2,3)=6
- d) "L\*1 =" . \$rezultat;

**55. Menționați metodele care fac parte din obiectul String pentru JavaScript?**

- a) toLowerCase();
- b) setTimeout();
- c) focus();
- d) strike();

**UNIVERSITATEA TITU MAIORESCU  
FACULTATEA DE INFORMATICĂ**

**MODELE ÎNTREBĂRI EXAMEN LICENȚĂ**

**Disciplina Comerț electronic**

1. Testarea site-urilor de comert electronic se poate face cu aplicatia software:

- a) eValid
- b) Rational SiteCheck
- c) SilkPerformer
- d) LoadRunner
- e) toate

2. Cardurile bancare nu contin:

- a) Numele emitentului, numarul cardului, perioada de valabilitate, numele posesorului-inscriptionate pe avers prin tiparire, gofrare sau gravare cu laser
- b) Sigla emitentului, holograma de securitate, logo-ul organizatiei, pe avers
- c) Suma de bani aflata in contul posesorului
- d) Banda magnetica si/sau smart card

3. Mecanismele de baza ale comertului electronic sunt:

- a) C2C
- b) B2C
- c) B2B
- d) toate

4. Conform Legii comertului electronic din Romania, falsificarea instrumentelor de plata electronica se pedepseste cu:

- a) in lege nu sunt prevazute sanctiuni
- b) cu inchisoare de la 3 la 12 ani si interzicerea unor drepturi
- c) cu amenda contraventionala sau inchisoare de la 1 la 2 ani
- d) tentativa se pedepseste

5. Proiectarea, implementarea si dezvoltarea unui site de comert electronic se poate face folosind aplicatia software:

- a) Pay Pal
- b) IBM BMP Suite
- c) MOODLE
- d) OsCommerce

6. Care din urmatoarele enunturi referitoare la comertul electronic este fals:

- a) eBay.com este cel mai mare magazin de comert electronic la nivel mondial
- b) Skimming-ul si phishing-ul utilizate in fraude cu carduri bancare si furt de identitate reprezinta infractiuni criminale informatice
- c) Protocolul SET (Secure Electronic Transaction) reprezinta un pachet de specificatii de securitate si de formate, dezvoltat de Visa si MasterCard, care permit utilizatorilor sa efectueze plati electronice cu cardurile bancare in Internet
- d) E-commerce (comertul electronic) reprezinta o strategie moderna in ceea ce priveste modul de a face afaceri, cea mai importanta componenta a sa fiind E-business (alaceri electronice)

7. Conform legislatiei nationale in vigoare, consumatorul are dreptul de a denunta unilateral contractul la distanta, fara penalitati si fara invocarea vreunui motiv in timp de:

- a) 30 de zile calendaristice de la semnare
- b) 15 zile lucratoare de la semnare
- c) 10 zile lucratoare de la semnare, daca comerciantul a specificat conditiile de denuntare unilaterala a contractului incheiat la distanta
- d) 90 de zile de la semnarea contractului incheiat la distanta

8. Arhitectura Java Card nu contine nivelul:

- a) Cardlet-uri (applet-uri), aplicatii care sunt incarcate in card printr-un incarcator standard (de exemplu, EZ Formatter);
- b) Masina Virtuala Java, inclusa in card, care este un interpretor care translateaza codul Java in instructiuni ale microprocesorului
- c) Sesiune de prezentare, nivel intermediar intre Cardlet-uri si Masina Virtuala Java
- d) Sistemul de operare GPOS- General Purpose Operating System

9. Sistemul electronic de transfer de fonduri (EFT-Electronic Found Transfer) reprezinta:

- a) ansamblul dispozitivelor si procedeele speciale care asigura deplasarea fluxurilor monetare de la platitor la beneficiar intr-un mediu exclusiv electronic
- b) un protocol de comunicatie între POS (Point Of Sales) si ATM (Automatic Teller Machine)
- c) o instructiune comuna la nivel mondial pentru sistemele bancare
- d) nici unul

10. Operatiunile cu carduri bancare cuprind urmatoarele etape:

- a) emiterea cardurilor
- b) acceptarea cardurilor
- c) distrugerea cardurilor
- d) toate

11. Dupa modul de stocare a caracteristicilor de securitate cardurile se impart in:

- a) carduri cu banda magnetica - care contin pe 3 piste ce memoreaza informatii criptate despre utilizator, emitent, algoritmul de codare etc;
- b) carduri cu microprocesor – smart-carduri- avand stocate in chip datele de securitate si permitand functionarea ca portofel electronic
- c) carduri de debit - care asigura utilizatorului achizitionarea de bunuri si servicii sau retrageri de numerar, cu conditia prezervarii unor fonduri intr-un cont de card, si efectuarea de cheltuieli in limita soldului disponibil
- d) numai a) si b)

12. Protocolul SET (Secure Electronic Transaction) asigură urmatoarele:

- a) un canal securizat între toate părțile implicate într-o tranzactie
- b) imbunătățirea secretului tranzactiilor; informatia este accesibilă părților unei tranzactii numai în măsura în care este necesar.
- c) autentificarea participantilor la tranzactii prin folosirea certificatelor digitale
- d) toate

13. In domeniul educatiei de tip e-learning, parte componenta a e-business, se poate implementa si utiliza platforma opensource:

- a) Zen Cart
- b) OSCommerce
- c) MOODLE
- d) IBM WebSphere

14. Care din urmatoarele enunturi este fals referitor la dezavantajele utilizarii e-commerce:

- a) aspectele privind securitatea comertului electronic – cea mai complexa problema
- b) acceptarea noilor modalitati de plata (bani electronici/digitali)
- c) intermediarii pot fi eliminati din lantul de aprovizionare
- d) existenta unui cadrul legal si normativ unitar la nivel global: cadrul fiscal, drepturile asupra proprietatii, protectia datelor consumatorului

15. Care dintre urmatoarele enunturi reprezinta componente ale e-business:

- a) CRM – Customer Relationship Management
- b) SCM – Supply Chain Management
- c) ERP – Enterprise Resource Planning
- d) toate

16. Un sistem electronic cu microplati permite efectuare de plati electronice cu valori de pana la:

- a) 1 euro
- b) 10 euro
- c) 100 euro
- d) 1000 euro

17. Care din urmatoarele modele nu este specific pentru e-commerce:

- a) Modelul Paul Timmers
- b) Modelul David Kosiur
- c) Modelul William Stallings
- d) Modelul Michael Rappa

18. Modelul David Kosiur prezinta cinci procese de baza care au loc in cadrul comertului electronic. Aceste procese sunt:

- a) distribuirea informatiilor, lansarea comenzilor, plata, onorarea comenzilor, service-ul si asistenta
- b) ofertarea, plata, onorarea comenzilor, service-ul si asistenta, postgarantie
- c) stabilirea pretului, lansarea produselor, plata, distribuitia, receptia
- d) distribuirea informatiilor, stabilirea pretului, livrarea, receptia, service-ul si asistenta

19. Care din elementele enumerate nu este specific unui sistem de comert electronic:

- a) serverele organizatiei care ofera servicii on-line accesibile prin Internet
- b) clientii (persoane fizice sau juridice) care se conecteaza la server folosind diverse tipuri de dispozitive
- c) conturile bancare ale participantilor la care sunt asociate carduri bancare (credit, debit)
- d) tranzactiile - raportate la bunurile si serviciile cumparate

20. In cadrul unui sistem de comert electronic o protectie adecvata trebuie acordata consumatorului, protectie care se refera la mai multe aspecte. Care dintre aspectele enumerate este fals?

- a) interzicerea publicitatii inselatoare
- b) interzicerea promovarii produselor
- c) reglementarea serviciilor financiare
- d) reguli privind raspunderea referitoare la produsele necorespunzatoare

21. Care din urmatoarele enunturi nu este un sistem de microplata electronica:

- a) Cybercoin (proiect CyberCash)
- b) Euro-MicroPayment (proiect UE)
- c) Millicent (proiect Digital)
- d) Micro-Payment (proiect Consortiul W3)

22. Care din urmatoarele protocoale este specific este specific pentru tranzactiile electronice:

- a) FTP
- b) ARP
- c) STP
- d) iKP

23. Conform protocolului 3KP toti participantii la o tranzactie electronica trebuie sa posede:

- a) bani in cont pentru a face tranzactia
- b) **certificat digital cu cheia sa publica, precum si cheia privata**
- c) doar un certificat digital
- d) canale de comunicatie securizate peste Internet

24. Atacurile informatice de tip *pharming* vizeaza compromiterea:

- a) serverelor de email
- b) serverelor de DHCP
- c) serverelor de Active Directory/Controler de domeniu
- d) **serverelor de DNS**

25. Un sistem electronic de plati poate fi definit ca un ansamblu de tranzactii generate de:

- a) **conversia banilor numerar in bani electronici si invers**
- b) verificarea conturilor bancare asociate cumparatorului sau vanzatorului
- c) schimbul de mesaje criptate si semnate electronic intre participantii (vanzator, cumparator, banci)
- d) operatiile bancare rezultate pe durata efectuarii tranzactiilor electronice

26. Care dintre enunturile de mai jos nu constituie o infractiune cu carti de credit:

- a) SKIMMING
- b) PIMMING
- c) **PHARMING**
- d) PHISING

27. Infrastructura de comunicatii a unei organizatii care realizeaza comert electronic trebuie sa contina urmatoarele doua elemente de baza:

- a) e-commerce front-end
- b) e-commerce back-end
- c) componente middleware
- d) **toate**

28. Ce etape implica realizarea unui site Web de comert electronic:

- a) analiza si planificarea site-ului
- b) proiectarea si testarea site-ului
- c) implementarea si promovarea
- d) **toate**

29. Care dintre enunturile de mai jos nu este un participant in tranzactiile de plata din protocoalele iKP:

- a) **verificator**
- b) cumparator
- c) achizitor
- d) vanzator

30. Protocolul SET defineste o diversitate de tranzactii care folosesc concepte criptografice. Care dintre conceptele enumerate mai jos nu se foloseste in protocolul SET:

- a) certificate digitale
- b) **SSL / TLS**
- c) semnatura duala
- d) semnatura digitala

31. Modelul unui sistem electronic de plati (Electronic Payment System-EPS) implica, in general, urmatoarele cel putin urmatoarele entitati care interactioneaza:

- a) o banca B, un cumparator C si un vanzator V
- b) o banca B, un procesator de carduri P, un cumparator C si un vanzator V
- c) un procesator de carduri P, un cumparator C si un vanzator V
- d) un cumparator C, o banca a lui C, un vanzator V si o banca a lui V

32. Tranzactiile electronice reprezinta schimburile de mesaje, sub forma unor protocoale, care se desfasoara intre entitatile ce joaca diverse roluri implicate intr-un EPS. Care dintre enunturile de mai jos nu este o tranzactie tipica intr-un:

- a) tranzactia de identificare a utilizatorilor
- b) tranzactia de retragere
- d) tranzactia de actualizare a platilor
- d) tranzactia de clearing

33. Un <scrip> din cadrul protocolului MilliCent are urmatoarele proprietati principale:

- a) are valoare doar pentru anumiti vanzatori
- b) se poate cheltui o singura data
- c) este rezistent la falsificare si greu de contrafacut
- d) toate

34. In domeniul platilor prin card, transferul electronic de fonduri poate fi realizat cu ajutorul:

- a) ghiseul automat de banca (ATM - Automatic Teller Machine)
- b) puncte electronice de vanzare la comercianti (EFT POS-Point of Sale)
- c) carduri bancare (cu banda magnetica sau smartcard)
- d) toate

35. La fiecare noua tranzactie, portofelul electronic vWALLET implementat de firma VeriFone recunoaste urmatoarele tipuri de instrumente:

- a) VISA, MasterCard
- b) VISA, MasterCard, American Express, JCB
- c) JCB
- d) American Express

36. Care dintre enunturile de mai jos nu reprezinta o norma in domeniul comertului electronic:

- a) UNCITRAL
- b) Directiva 31/2000/CE
- c) Legea nr.1/2011
- d) Legea 365/2002

37. Una dintre afirmatiile de mai jos referitoare la aspectele critice si dezavantajele comertului electronic este falsa. Care este aceea?

- a) aspectele privind securitatea comertului electronic – cea mai complexa problema
- b) acceptarea noilor modalitati de plata (bani electronici/digitali)
- c) stocurile pot fi minimizate sau chiar eliminate prin procese de productie „Just-In-Time”
- d) cadrul legal si normativ: cadrul fiscal, drepturile asupra proprietatii, protectia datelor consumatorului

38. In raport cu sursa de acoperire a cheltuielilor cardurile se impart in:

- a) carduri cu banda magnetica - care contin pe 3 piste ce memoreaza informatii criptate despre utilizator, emitent, algoritmul de codare etc.
- b) carduri de credit - care asigura utilizatorului achizitionarea de bunuri si servicii sau retrageri de numerar, pe baza unei linii de credit acordate posesorului de card
- c) carduri de debit - care asigura utilizatorului achizitionarea de bunuri si servicii sau retrageri de numerar, cu conditia prezervarii unor fonduri intr-un cont de card, si efectuarea de cheltuieli in limita soldului disponibil



d) numai b) si c)

39. Cardurile cu microprocesor se impart in urmatoarele categorii:

- a) carduri cu contact (contact-card)
- b) carduri cu memorie (memory-card)
- c) carduri fara contact (contactless-card)
- d) numai a) si c)

40. Marcati cu A-adevărat sau F-fals următoarele enunturi referitoare la securitatea e-commerce:

- a) atacurile de tip DoS (Denial of Service – Refuzul serviciului) afectează funcția de disponibilitate a informației; \_\_\_\_\_ A
- b) un craker este un utilizator Internet care nu cauzează pierderi sau pagube prin acțiunile sale; \_\_\_\_\_ F
- c) printre cei mai cunoscuți algoritmi de cifrare asimetrică sunt AES, DES, CAST, IDEA; \_\_\_\_\_ F
- d) elementele de bază ale PKI (Infrastructuri de Chei Publice) sunt certificatele digitale, autoritățile de certificare și serviciile de management al certificatelor; \_\_\_\_\_ A

41. Care din următoarele enunturi nu reprezintă un tip de organizație de comerț electronic:

- a) organizații de tip brick-and-mortar
- b) organizații de tip click-and-brick sau click-and-mortar
- c) organizații de tip click-and-click
- d) organizații de tip virtual business

42. Conform OG 34/2006, achizițiile publice în România se derulează pe platforma online:

- a) SEAP
- b) SETI
- c) ANISP
- d) ORDA

43. Tranzacțiile electronice din punctul de vedere al comerțului electronic prezintă următoarele cerințe:

- a) cerințe privind legislația aplicabilă
- b) cerințe privind standardele acceptate
- c) cerințe privind securitatea tranzacțiilor
- d) toate

44. Utilizarea cardurilor bancare la retragerea de numerar dintr-un ATM presupune autentificarea cu:

- a) un factor sau A (ce am? sau ce știu?)
- b) doi factori sau AA (ce am ? și ce știu?)
- c) trei factori sau AAA (ce am ? și ce știu? și cine sunt?)
- d) patru factori sau AAAA (ce am ? și ce știu? și cine sunt? și unde sunt?)

45. Principalele amenințări asupra tranzacțiilor din comerțul electronic sunt:

- a) Interceptarea informațiilor referitoare la plată (de exemplu numărul cardului de credit și a numărului CVV)
- b) Interceptarea parolelor sau a altor informații de acces ale sistemului (de exemplu, PIN)
- c) Modificarea datelor unei tranzacții înainte de livrare
- d) toate

46. Un sistem electronic de plati (Electronic Payment System-EPS) poate fi definit ca un ansamblu de tranzactii generate de:

- a) conversia banilor numerar (cash sau din cont) in bani electronici si invers
- b) transferul banilor electronici intre utilizatorii care folosesc sistemul EPS
- c) extragerea de bani din cont, efectuarea unei plati sau depunerea banilor electronici într-o banca
- d) toate

47. Care din urmatoarele roluri de mai jos nu este asociat unei Autorități de Certificare (AC):

- a) să emită si să revoce certificate digitale X.509 v3
- b) să efectueze tranzactii de bani electronici
- c) să autentifice cererile de chei publice ale utilizatorilor
- d) să înlocuiască certificate digitale X.509 v3

48. Un portofel electronic poate opera in următoarele moduri:

- a) Inchis / Deschis - cand acelasi operator emite carduri si acceptă tranzactii sau când se face compensarea tranzactiilor între diferiti emitenti
- b) Pre-platit/Pre-autorizat. Un portofel electronic pre-platit inglobeaza (sub forma electronica) bani reali, transferati dintr-un cont bancar sau dintr-un depozit de numerar. Un portofel electronic pre-autorizat inglobeaza valoarea (costul) unei cumparaturi, pe care operatorul portofelului o garanteaza (se bazeaza pe o linie de credit)
- c) De unica folosinta / Reincarcabil
- d) toate

49. Indicatorul Major de Industrie – MII (Major Industry Identifier) pentru VISA si MasterCard au valorile:

- a) 2 si 3
- b) 3 si 4
- c) 4 si 5
- d) 5 si 6

50. Un certificat digital emis de o Autoritate de Certificare nu contine:

- a) cheia secreta a detinatorului
- b) cheia publica a detinatorului
- c) Autoritatea de Certificare care a emis certiificatul digital
- d) algoritmii folositi (RSA cu MD5)