G Ri Le Bp C - grule examen BPC

Bazele programării calculatoarelor Basics of Programming (Academia de Studii Economice din Bucure□ti)

GRILE BPC

1. Reprezentarea prin pseudocod este permisa numai pentru structurile:

1.BLOCK;
2.IF-THEN-ELSE,
3.CASE-OF,
4.WHILE-DO,
5.DO-UNTIL,
6.DO-FOR.

a. 1,2,3,4,5 si 6
b. 1,2,3,4 si 5
c. 2,3,4,5 si 6
d. 1,2 si 4
e. 1,2 si 5

INSTRUCTIUNI DE RAMIFICARE: IF-THEN-ELSE; IF-THEN; CASE-OF
INSTRUCTIUNI REPETITIVE: WHILE-DO, DO-UNTIL, DO-FOR

2. Fie o matrice Anxm. Sa se stabileasca ce realizeaza secventa urmatoare: g=0; for(i=0;i<n-1;i++) for(j=i+1; j<n,j++) if (a[i][j]!=0) g=1 (este triunghiulara superior daca elementele aflate deasupra diagonalei principale sunt diferite de 0) .

a. Verifica daca matricea e nula
b. Verifica daca matricea este triunghiulara superior
c. Verifica daca matricea are numai elemente diferite de zero
d. Verifica daca matricea este triunghiulara inferior
e. Verifica daca triunghiul superior format de cele doua diagonale sunt elementediferite de zero

3. Functia rang pentru un element A(i,j) dintr-o matrice Am,n **memorata invers lexicografic** este data de relatia:

a. m*(i-1)+j
b. n*(i-1)+j
c. m*(j-1)+i
d. n*(j-1)+i
e. n*m(i-1)+j

4. Functia rang pentru un element A(i,j) dintr-o matrice An,m **memorata lexicografic** estedata de relatia:

a. m*(i-1)+j
b. n*(i-1)+j
c. m*(j-1)+i
d. n*(j-1)+i
e. n*m(i-1)+j

5. Functia rang al unui element a(i,j,k) dintr-un masiv tridimensional Am*n*p, **memorat invers lexicografic** este:

a. i+m(j-1+n(k-1))
b. j+m(k-1+n(i-1))
c. k+n(i-1+p(j-1))
d. i+n(k-1+p(j-1))
e. j+p(i-1+m(k-1))

6. Functia rang al unui element a(i,j,k) dintr-un masiv tridimensional Am*n*p, **memorat lexicografic**, este:
a. i+p(j-1+m(k-1))
b. k+n(j-1+p(i-1))
c. k+n(i-1+p(j-1))
d. i+n(k-1+p(j-1))
e. j+p(i-1+m(k-1))

7. Metoda dublarii codurilor:
1) se foloseste la structurarea alternativelor sau repetitivelor;
2)conta in dublarea, ori de cate ori este nevoie, a unui cod(a unei actiuni sau conditii), astfel incat sa se obtina numai structuri fundamentale;
3) se foloseste numai pentru structurarea alternativelor;
4) se introduce o variabila booleana
5) se foloseste numai pentru structurarea repetitivelor

a. 2,3
b. 2,4,5
c. 2,3,4
d. 1,2,4
e. 1,2

7.5 Metoda introducerii unei variabile boolene:

1) se foloseste la structurarea alternativelor sau repetitivelor;
2)conta in dublarea, ori de cate ori este nevoie, a unui cod(a unei actiuni sau conditii), astfel incat sa se obtina numai structuri fundamentale;
3) se foloseste numai pentru structurarea alternativelor;
4) se introduce o variabila booleana
5) se foloseste numai pentru structurarea repetitivelor

Raspunsul este 4 si 5

8. Un operand poate fi una din urmatoarele constructii:
1) o constanta simbolica
2) un literal
3) o variabila simpla
4) numele unui masiv
5) numele unui tip de data
6) numele unei functii
7) referire a unui element de masiv
8) referirea unui camp de articol
9) apelul unei functii
10) o expresie

a. 3,5,7,8,10
b. 1,2,3,5,7,8
c. 1,2,3,4,5,6,7,8,10
d. Toate
e. 1,2,3,4,5,7,8,9

10. In operatia de atribuire(v=e), e poate fi:
1. variabila simpla,
2. Element de masiv,
3. expresie aritmetica,

4. Expresie relationala,
5. Expresie logica.

a) Toate b) 1,2,3 si 4 c) 1,2 si 4 d) 1,3 si 5
e) 1,2,3 si 5

11. Operatia de citire desemneaza:
a. Preluarea datelor de la tastatura
b. Preluarea datelor de pe suporti magnetici
c. Transferul datelor intre zone de memorie principala
d. Transferul datelor de pe suporti externi in memoria principala
e. Transferul datelor in buffer

12. Operatia de scriere desemneaza:
a. afisarea datelor pe monitor
b. scrierea datelor pe suporti magnetici
c. transferul datelor intre zone de memorie principal
d. Transferul datelor din memoria principal pe suporti externi
e. transferal datelor in buffer

//arc etichetat cu informatia, inscrierea pe suportul extern a valorilor memorate in locatiile de memorie corespunzatoare unor variabile/ transferul datelor din MP in exteriorul sistemului de calcul, cu sau fara conversie.

13. Un algoritm iterativ este:
a. Un algoritm care se autoapeleaza
b. Un proces repetitiv static
c. Un proces repetitiv dinamic
d. Un proces repetitiv prin care valoarea unei variabile nu depinde de valorile ei anterioare
e. Un proces alternativ prin care valoarea unei variabile nu depinde de valorile ei anterioare
(iterativ- nu depinde de valorile anterioare)

14. Un algoritm recursiv este:
a. Un algoritm care se autoapeleaza
b. Un proces repetitiv static
c. Un proces repetitiv dinamic
d. Un proces repetitiv prin care valoarea unei variabile se determina pe baza a cel putin uneia dintre valorile ei anterioare
e. Un proces alternativ prin care valoarea unei variabile se determina pe baza a cel putin uneia dintre valorile ei anterioare
(recursiv-depinde de anumite valori interioare)

15. Iterativitatea este:
a) un proces prin care rezultatul este obţinut ca urmare a execuţiei repetate a unui set de operaţii, de fiecare dată cu aceleaşi valori de intrare;
b) un proces repetitiv static;
c) un proces repetitiv dinamic;
d) un proces repetitiv prin care valoarea unei variabile se determină pe baza unei valori anterioare;
e) un proces prin care rezultatul este obţinut ca urmare a execuţiei repetate a unui set de operaţii, de fiecare dată cu alte valori de intrare.

16. Recursivitatea este:
a) un algoritm care apelează un alt algoritm;
b) un proces repetitiv static;
c) un proces repetitiv dinamic;

d) un proces iterativ prin care valoarea unei variabile se determină pe baza a cel puţin uneia dintre valorile ei anterioare;
e) un proces alternativ prin care valoarea unei variabile se determină pe baza a cel puţin uneia dintre valorile ei anterioare.

17. Codurile ASCII grafice sunt:
a) 0-31
b) 0-32
c) 32-127
d) 32-128
e) 128-255

18. Codurile ASCII ale caracterelor direct afisabile apartin intervalului:
a. 0-31
b. 0-32
c. 32-127
d. 32-128
e. 128-255

19. Codurile ASCII ale cifrelor de la 0 la 9 apartin intervalului :
a. 0-31
b. 0-32
c. 32-127
d. 32-128
e. 128-255

20. Codurile ASCII de control apartin intervalului:
a. 0-31
b. 0-32
c. 32-127
d. 32-128
e. 128-255

21. Lista este o structura:
a. complementara
b. omogena cu acces secvential
c. omogena cu acces direct
d. eterogena cu acces secvential
e. eterogena cu acces direct

22. Coada este o lista la care:
a) Inserarea si stergerea se fac in capul listei si citirea se face la baza listei
b) Inserarea, stergerea si citirea se fac in capul listei
c) Inserarea, stergerea si citirea se fac la baza listei
d) Inserarea se face la baza listei, iar stergerea si citirea se fac la capul listei
e) Inserarea si stergerea se fac la baza listei si citirea se face in capul listei

23. Stiva este o lista la care:
a) Inserarea si stergerea se fac in capul listei si citirea se face la baza listei
b) Inserarea, stergerea si citirea se fac in capul listei
c) Inserarea, stergerea si citirea se fac la baza listei
d) Inserarea se face la baza listei, iar stergerea si citirea se fac la capul listei
e) Inserarea si stergerea se fac la baza listei si citirea se face in capul listei

24. Masivul este o structura:
a. complementara
b. omogena cu acces secvential
c. omogena cu acces direct
d. eterogena cu acces secvential
e. eterogena cu acces direct

25. Articolul este o structura:
a. complementara
b. omogena cu acces secvential
c. omogena cu acces direct
d. eterogena cu acces secvential
e. eterogena cu acces direct

26. Structurile privilegiate sunt:
1.BLOCK(s1,s2), 2.IF-THEN-ELSE(c,s1,s2), 3.IF-THEN(c,s), 4.CASE-OF(i,s1,s2,…,sn,s), 5.WHILE-DO(c,s), 6.DO-UNTIL(s,c), 7.DO-FOR(v,vi,vf,vr,s)
a. 1,2,3,4,5,6,7
b. 1,2,3,5,6
c. 1,2,5,6,7
d. 1,2,5
e. 1,2,6

27. O data este definita de urmatorii parametrii:
1. Identificator, 2. Consistenta, 3. Valoare, 4. Atribute, 5. Semnificatie, 6. Paradigma
a) Toti
b) 1,3,4 si 5
c) 1,2,3,4 si 6
d) 1,3,5 si 6
e) 1,3 si 4

28. O data reprezentata VF algebrica pe 2o are valoarea minima:
a. -2^{16}
b. -2^{15}
c. $-2^{16}+1$
d. $-2^{15}+1$
e. 0

O DATA reprezentata pe VF ALGEBRICA pe 2 octeti are valoarea maxima: $2^{15}-1$ (1 octet: $2^{7}-1$, 4 octeti: $2^{31}-1$).

O DATA reprezentata pe VF ALGEBRICA pe 2 octeti are valoarea minima: -2^{15}

(1 octet:-2^{7}, 4 octeti: -2^{31}).

O DATA reprezentata pe VF ARITMETICA pe 1/2 octeti are valoarea maxima: $2^{8}-1$ / $2^{16}-1$ (nr nat, fara semn) (si valoarea minima este 0)

29. O data reprezentata VF aritmetica pe 2o are valoarea maxima:
a. 2^{16}
b. 2^{15}
c. 2^{16}-1
d. 2^{15}-1
e. 0

30. Numarul de iteratii intr-o structura DO-FOR(v,vi,vf,vr,s) se determina dupa relatia
a. |vf-vi|/vr+1 (for i=0; i<9 , i++)
b. |(vf-vi)/vr|+1 **Vi Vf Vr s-conditie/instructiune**
c. [vf-vi/vr]+1
d. [|vf-vi|/vr]+1
e. [|vi-vf\/vr]+1

31. Numarul de iteratii intr-o structura WHILE-DO(vi<=vf,s) este :
a. [(vf-vi)/vr]+1
b. 0
c. Nedeterminat
d. [vi-vf]+1
e. 1
32. Deplasarea campului ck, de lungime lk, dintr-un articol este data de relatia:
a. D(ck)=D(ck-1)+lk-1
b. ∑D(ck)=D(ck)
c. D(ck)=D(c1)+(k-1)*lk
d. D(ck)=(k-1)*lk e. ∑lk

33. Constantele simbolice sunt:
a. Secventele text care pot fi proiectate si realizate independent
b. Siruri de zero sau mai multe caractere, delimitate prin ghilimele
c. Variabile initializate la declarare, pentru care se rezerva memorie, dar continutul lor nu poate fi modificat pe parcursul executiei programului
d. Multimi finite de elemente omogene
e. Literari carora li se asociaza identificatori

34. Constantele obiect sunt:
a. Secventele text care pot fi proiectate si realizate independent
b. Siruri de zero sau mai multe caractere, delimitate prin ghilimele
c. Variabile initializate la declarare, pentru care se rezerva memorie, dar continutul lor nu poate fi modificat pe parcursul executiei programului
d. Multimi finite de elemente omogene
e. Literari carora li se asociaza identificatori

35. Un cod ASCII este reprezentat:
a) Virgula fixa aritmetica pe 1 octet
b) Virgula fixa algebrica pe 1 octet
c) Virgula mobila pe 1 octet
d) Cod invers
e) Cod complementar

36. Conceptele principale cristalizate in domeniul programarii structurate sunt:
1) proiectarea top-down;
2) proiectarea modulara;
3)proiectarea structurata;
4) proiectarea distribuita;
5) proiectarea orientata obiect

a. 1,3 b. 2,3 c. toate **d. 1,2,3** e. 1,3,4,5

37. Structura de date se defineste ca:
a. O colectie de date pe care s-a definit un mecanism de selectare a componentelor
b. O colectie de date la care o componenta este independenta de celelalte

c. O colectie de date compusa din subcolectii de acelasi tip
d. O colectie de date compusa din subcolectii de tipuri diferite
e. O colectie recursiva de date

38. Se numeste schema logica un graf orientat in care:
1) Exista un singur bloc START
2) orice arc este etichetat cu una din urmatoarele informatii: START sau STOP; o citire sau o scriere; o atribuire; un predicat, in caare caz extremitatea initiala a arcului este extremitatea initiala a unui bloc de ramificatie
3)orice arc face parte din cel putin un drum care incepe in blocul START si se termina in blocul STOP
4) exista un singur bloc STOP
5) exista un singur blovc START si mai multe blocuri STOP

a. Toate b. 1,2,4 c. 2,3 d. 2,3,5 **e. 1,2,3,4**

39. Numarul real, in zecimal, a carui reprezentare interna in binar este 1110001,011 este:
a. -49,2
b. 49,3
c. 113,3
d. 113,375
e. -113,375

40. Numarul in zecimal al carui reprezentare interna in VF aritmetica este 10001111 este:
a) 143
b) -15
c) 103
d) -103
e) 25

41. Numărul în zecimal a cărui reprezentare internă în VF algebrică este 10001111 este:
a) –113;
b) 143;
c) –143;
d) –15;
e) 113.
10001111-in CC
10001110-in CI
11110001-in CD
2^0+2^4+2^5+2^6= -(1+ 16+32+64)= -113

42. Blocurile dintr-o subschemă logică sunt etichetate cu una din informaţiile:
1)START;
2)citire;
3)scriere;
4)expresie aritmetică;
5)expresie logică;
6)expresie relaţională;
7)sir de caractere;
8)atribuire;
9)salt necondiţionat;
10)STOP.

a)1,2,3,5,6,7,8 sau 10;
b)1,2,3,5,6,8 sau 10;
c)1,2,3,4,8 sau 10;

d)1,2,3,4,6,8,9 sau 10;
e)oricare

43. Dintre secvenţele următoare sunt corecte numai:
1)IF-THEN-ELSE(c,F,F);
2)IF-THEN(c,F);
3)BLOCK(F);
4)WHILE-DO(c,F);
5)DO-UNTIL(F,c);
6)DO-FOR(v,vi,vf,vr,F).

a)1,2,3,4 şi 5;
b) 1,2 şi 4;
c)1,2 şi 5;
d)niciuna;
e)toate.

44. Fazele dezvoltării programelor sunt:
1) editare; 2) verificare sintaxă; 3) compilare; 4) editare legături; 5) lansare în execuţie; 6) testare.

a) 1,3,4 şi 5;
b) 1,2,3,4 şi 5;
c) 1,3,4,5 şi 6;
d) 1,2,3 şi 4;
e) toate.

45. Secvenţa: i=0; while ((i<n) && (x[i] != a)) i++; calculează:
a) ultima apariţie a valorii date a într-un vector;
b) eliminarea primei apariţii a valorii date a într-un vector;
c) ultima apariţie a unei valori diferite de valoarea dată a;
d) prima apariţie a valorii date a într-un vector;
e) prima apariţie în vector a unei valori diferite de valoarea dată a.

46. Fie o matrice Amxm. Să se stabilească ce calculează secvenţa următoare: p=a[0][1]; for(i=0; i<m; i++) for (j=i; j<m; j++) if (a[i][j]<p) p=a[i][j];
a) maximul din triunghiul de deasupra diagonalei principale (exclusiv diagonala);
b) maximul din triunghiul de sub diagonala principală (exclusiv diagonala);
c) maximul din triunghiul inferior format de cele două diagonale (inclusiv diagonalele);
d) minimul din triunghiul de deasupra diagonalei principale (inclusiv diagonala);
e) minimul din triunghiul de deasupra diagonalei principale (exclusiv diagonala)

47. Asociaţi fiecărui punct una dintre caracteristicile principale ale unui algoritm:

a) algoritmul trebuie să prevadă modul de soluţionare a tuturor situaţiilor care pot apărea în rezolvarea problemei respective, într-o manieră fără ambiguităţi sau neclarităţi: claritate;
b) un algoritm nu trebuie conceput pentru o problemă particulară, ci pentru o clasă generală de problem: generalitate;
c) operaţiile trebuie astfel concepute încât algoritmul să se termine într-un număr finit de paşi, cunoscut sau necunoscut: finitudine;
(Ordinea generala este: Generalitate , claritate , finitudine)
Aici este b) , a) , c)
48.Caracteristicile fiecarui algoritm: generalitate, claritate, finitudine

49. Reprezentarea prin arbori este permisa numai prin structurile

1. BLOCK
2. IF-THEN-ELSE
3. CASE-OF
4. WHILE-DO
5. DO-UNTIL
6. DO-FOR

a. TOATE
b. 1,2,3,4 si 5
c. 2,3,4,5 si 6
d. 1,2 si 4
e. 1,2 si 5

50. Care dintre urmatoarele secvente **nu** realizeaza suma a n elemente ale unui vector:
a. S=0; for(i=0;i<n;i++) s+=x[i];
b. S=0; for(i=n-1;i>=0;i--) s+=x[i];
c. S=0; i=0; while(i<n) {s+=x[i];i++}
d. S=0; i=n-1; while(i>0) {s+=x[i];i--}
e. S=0; i=0; do{s+=x[i];i++} while (i<n)
51. Care din următoarele secvenţe realizează suma a n elemente ale unui vector:
1. s=0; for(i=0; i<n; i++) s+=x[i];
2. s=0; for(i=n-1; i>=0; i--) s+=x[i];
3. s=0; i=0; while (i<n) {s+=x[i]; i++;} ;
4. s=0; i=n-1; while (i>0) {s+=x[i]; i--;} ;
5. s=0; i=0; do { s+=x[i]; i++; } while(i<n);
a); 1,2,4,5; b)1,3,4,5; **c)1,2,3,5**; d)2,3,4,5; e) niciun răspuns din cele prezentate.

52. Secventa:
```
for (i = 0; i < n - 1; i++)
{z = x[i]; p = i;
for (j = i + 1; j < n; j++)
        {if (x[j] < z)
                {z = x[j]; p = j;}
        a = x[i]; x[i] = z; x[p] = a;}
}
```

a. Minimul dintr-un vector cu retinerea pozitiei primei aparitii
b. Minimul dintr-un vector cu retinerea pozitiei ultimei aparitii
c. Sortarea unui vector prin metoda bulelor
d. Sortarea unui vector prin metoda selectiei
e. Cautarea unei valori date dintr-un vector

53. Schema logică structurată (s.l.s.) se defineşte astfel:
a) Blocurile START, STOP, de intrare/ieşire şi de atribuire sunt scheme logice structurate;
b) Subscheme: structură secvenţială, structura alternativă IF-THEN-ELSE, structura repetitivă WHILE-DO – sunt subcheme logice structurate
c) Numai subschemele: structură secvenţială, structura alternativă IF-THEN-ELSE - sunt subcheme logice structurate
d) Orice s.l.s. se obţine plecând de la (a) şi aplicând de un număr finit de ori regulile (b)
e) Orice s.l.s. se obţine plecând de la (a) şi aplicând de un număr finit de ori regulile (c)

54. Urmatoarele secvente realizeaza suma elementelor de rang impar dintr-un vector:
1. s=0; i=0; while (i<n) {s+=x[i]; i+=2;
2. s=0; i=n-1; while(i>=0) {s+=x[i]; i-=2;
3. int n; s=0; for(i=0; i<(n/2); i++} s=s+x[2*i];

4. int n; s=0; for(i=0; i<((n/2)+1); i++) s=s+x[2i]
*
a) 1
b) 1 si 3
c) 1 si 4
d) 1, 2 si 3
e) Toate

55. Urmatoarele secvente descriu algoritmi recursivi:
1. s=0; for(i=n-1; i>0; i--) n+=n[i];
2. for(i=0; i<n; i++) y[i]=x[i];
3. nr=0; i=0; while(i<n) {if (x[i]=0 nr=1; i++;}
4. for(i=0; i<n; i++) z[i]=x[i]+y[i];
5. i=0; z=0; do {z+=x[i]*y[i]; i++} while(i<n)
6. S=1; for(i=0; i<n; i++) S*=i;

a) toate
b) 1,3,5,6
c) 2,4,6 d)3, 5 e) niciunul

56. Secventa p=0; for(i=0;i<m;i++) {j=0; while((j<n)&&(x[i]!=y[j]))j++; if(j==n) z[p++]=x[i];}
realizeaza:
a. Operatia de intersectie dintre doua multimi
b. Toate aparitiile unei valori date intr-un vector
c. Operatia de reuniune a doua multimi
d. Diferenta dintre multimea y si multimea x (anume y-x)
e. Diferenta dintre multimea x si multimea y (anume x-y)

57. Secventa: for(i=0;i<n;i++) for(j=i+1;j++) if(x[i]==x[j]) {for(k=j;k<n-1;k++) x[k]=x[k+1]; n--;j--;}
realizeaza:
a. Sortarea elementelor unui vector
b. Duplicarea elementelor consecvente dintr-un vector
c. Eliminarea tuturor aparitiilor egale cu primul element din vector
d. Compactarea elementelor unui vector
e. Inversarea elementelor din vector

58. Secventa:

```
a = x[0];
p = 0;
for(i=1;i<n;i++)
        if (x[i] >= a) {a = x[i];p = i;}
```

a. maximul dintr-un vector si prima sa aparitie;
b. ultima aparitie a unei valori mai mare decat prima valoare din vector;
c. maximul dintr-un vector si ultima sa aparitie;
d. ultima valoare din vector si pozitia acestei valori;
e. minimul dintr-un vector si ultima sa aparitie.

59. **Teorema de structura** = orice schema logica este echivalenta cu o schema logica structurata.

60. Specificati care va fi valoarea var c dupa realizarea urmatoarei secvente:

int a =8; b=10, c
c=(a>b)?a:b => se afiseaza valoarea lui b
Valoarea si tipul acestei expresii sunt identice cu cele ale expresie_2 (daca expresie_1 este adevarata)

Sau cele ale expresie_3 (daca expresie_1 este falsa)
daca (..) adevarata se afiseaza a
daca (..) falsa se afiseaza b

61. Structura DO-FOR(v,vi,vf,vr,s) este echivalenta cu:
a. BLOCK (v=vf, DO-UNTIL(BLOCK(s,v=v-vr), v<=vi))
b. BLOCK (v=vi, IF-THEN(v<=vf, DO-UNTIL(BLOCK(s,v=v+vr),v>vf)))
c. BLOCK (v=vf, WHILE-DO(v>vi,BLOCK(s,v=v-vr)))
d. BLOCK (v=vi, WHILE-DO(v<vf, BLOCK (s, v=v+vr)))
e. BLOCK (V=VI, DO UNTIL(BLOCK(v=v+vr,s), v>vf))

62. Un ALGORITM STRUCTURAT este echivalent cu un ALGORITM pus sub una din formele: BLOCK(s1,s2), IF-THEN-ELSE(c,s1,s2), WHILE-DO(c,s).

63. REPREZENTAREA PRIN ARBORI este permisa numai pt structurile BLOCK, IF-THEN-ELSE, IF-THEN, CASE-OF, WHILE-DO, DO-UNTIL, DO-FOR.

64. Specificaţi care va fi valoarea variabilei c, după realizarea următoarei secvenţe: int a=25, b=3, c; c=(a<b)?a:b; **a)3**; b) 1; c) 0; d) nici una din valorile menţionate; e) 25

65. Specificați care va fi valoarea variabilei d, după realizarea următoarei secvențe: int a=10, b=3, c, d; d=(b+=a, c=b*a, a/2); a) nici una din valorile menţionate; **b) 5**; c) 130; d) 13; e) 10;

66. Specificați care va fi valoarea variabilei c, după realizarea următoarei secvențe: char a, b, c; a = 9; b = 10; c = a ^ b; a) 8; b) 11; **c) 3**; d) 10; e) nici una din valorile menţionate;

67. Ce realizează secvența următoare, unde a este o variabilă de tip char: for (i = 0; i <= 7; i++){printf("%d", a & 1);a = a >> 1;} a) nici una din variantele menționate; **b) afișează în ordine inversă, biţii unui octet**; c) afişează, în ordine, biţii unui octet; d) afișează întotdeauna un șir de 8 valori de 1; e) afișează întotdeauna un șir de 8 valori de 0;

68. Specificați care va fi valoarea variabilei dim, după realizarea următoarei secvențe: char sir2[]="mama"; dim=sizeof(sir2); **a) 5** ; b) 4; c) nici una din valorile menţionate; d) 8; e) 10;

69. Fie o matrice A mxn. Să se stabilească ce realizează secvenţa următoare: p=0; for (i=0; i<m; i++) {j=1; while (j<n && a[i][j] == a[i][0]) j++; if (j<n) x[p++]=i;} a) determină elementele din matrice egale cu o valoare dată; b) determină coloanele dintr-o matrice care au elementele constante; c) determină liniile dintr-o matrice care au elementele constante; **d) determină liniile dintr-o matrice care nu au elemente constante**; e) determină toate coloanele care au elemente diferite de primul element.

70. Secvenţa: p=0; for (i=0;i<m;i++){j=0; while ((j<n) && (x[i] !=y[j])) j++; if (j==n) z[p++]=x[i];} realizează: a) operaţia de intersecţie dintre două mulţimi; b) toate aparițiile unei valori date într-un vector; c) operaţia de reuniune a două mulţimi; d) diferenţa dintre mulţimea y şi mulţimea x (anume y-x); **e) diferenţa dintre mulţimea x şi mulţimea y (anume x-y).**

71. Numărul în zecimal a cărui reprezentare internă în VF aritmetică este 10001010 este: : a) –118; **b) 138;** c) –138; d) –10; e) 118.

72. Numărul în zecimal a cărui reprezentare internă în VF algebrică este 10001010 este**: a) –118**; b)138; c) –138; d) –10; e) 118.

73. Structura WHILE-DO(c,s) este echivalenta cu:
a. DO-UNTIL(c,s)
b. BLOCK(s, DO-UNTIL(s,c))
c. IF-THEN(c, DO-UNTIL(s,c))
d. BLOCK (s, IF-THEN(c,s));
e. DO-UNTIL(IF-THEN(c,s),c)

74. Structura DO-UNTIL(s,c) este echivalenta cu:
a. WHILE-DO(c,s)
b. BLOCK (s, WHILE-DO(c,s))
c. IF-THEN(c, WHILE-DO(c,s))
d. BLOCK(s, IF-THEN(c, WHILE-DO(c,s))
e. IF-THEN(c,WHILE-DO(c,s))

75. Intr-o abordare schematic, enumerati etapele de realizare a sistemelor informatice.
1) Studiul si analiza sistemului informational actual;
2) Proiectarea de ansamblu;
3) Proiectarea de detaliu;
4) Elaborarea programelor
5) Implementarea si exploatarea sistemului

76. Specificaţi care va fi valoarea variabilei c, după realizarea următoarei secvenţe: int a=7, b=9, c;
c=(a>b)?a:b
Variabila c va fi 9.

77. Scrieti secventele echivalente pentru urmatoarele exemple:
y=x++; => y=x, x=x+1
y=--x; => x=x-1, y=x

78. Triunghiul de sub diagonal secundara (inclusive diagonala) unei matrice patrate se poate parcurge numai cu secventele:
1. for (i=0;i<n;i++) for (j=n-i-1,j<n;j++)…
2. for (i=0;i<n;i++) for (j=n-1,j<n-i-1;j--)…
3. for (i=n-1;i>=0;i--) for (j=n-i-1,j<n;j++)…
4. for (i=n-1;i>=0;i--) for (j=n-1,j<n-i-1;j--)…
5. for (j=0;j<n;j++) for (i=n-j-1,i<n;i++)…
6. for (j=0;j<n;j++) for (i=n-1,i>=n-j-1;i--)…
7. for (j=n-1;j>=0;j--) for (i=n-j-1,i<n;i++)…
8. for (j=n-1;j>=0;j--) for (i=n-1,i>=n-j-1;i--) do…

80. Urmatoarele secvente descriu un algoritm recursiv:
1. s=0; for (i=n-1, i>=0; i--) s+=x[i];
2. for (i=0;i<n;i++) y[i]=x[i]
3. nr=0; i=0; while (i<n) { if (x[i]>0) nr+=1; i++;
4. for (i=0;i<n;i++) z[i]=x[i]*y[i]
5. i=0; z=0; do {z+=x[i]*y[i]; i++}while (i<n)
6. s=1; for (i=0;i<n;i++) s*=1

81.
```
p = a[0][1];
for (i = 0; i < (m - 1) / 2; i++)
       for (j = i + 1; j < m - 1 - i; j++)
              if (a[i][j] > p)
                             p = a[i][j];
```

a) maximul din triunghiul de sub diagonala secundara (exclusiv diagonala);
b) maximul din triunghiul inferior format de cele două diagonale (inclusiv diagonalele);
c) minimul din triunghiul superior format de cele două diagonale (exclusiv diagonala);
d) maximul din triunghiul superior format de cele doua diagonale (exclusiv diagonala)
e) maximul din triunghiul de deasupra diagonala principală (exclusiv diagonal)

82. Specificati cum va arata secventa de cod urmatoare dupa preprocesare:
#define N 10
#define M 10
#define MAX (M+N)
#define DIM (a, b) (a)*(b)
Char v(N), v1[10+DIM(5+M,6)]
Char v1[10*MAX];
Char m[M][N]

Raspuns:

Char v(10), v1[10+(5+10)*6)]
Char v1[10*(10+10)];
Char m[10][10]

83.
Informatia de care un program are nevoie pentru a is indeplini obiectivul se numeste:

a) Contributie
b) Litera
c) Date de Intrare
d) Date de iesire
e) efort

84.

Datele simple sunt: 1) vectorii 2)enumeratie 3)char 4)struct 5)int:

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

85.

Structura care necesita repetarea pana cand o secventa este intalnita se numeste:

a) Secventa
b) Conditie
c) Bucla
d) Selectie
e) Alternativa

86

_____ul converteste limbajul procedural al programatorului intr-un limbaj masina:

a)compilator

b)analizator

c)designer

d)editor

e)convertor

87.

Structura de control folosita pentru a exectua o serie de instructiuni una dupa alta se numeste:

a) Intamplatoare
b) Bucla
c) Selectiva
d) Secventiala
e) Repetitiva

88.

Unealta de programare care foloseste simboluri legate pentru a arata o secventa de pasi care trebuie sa resolve un program se numeste:

A) Algoritmul
B) Pseudocod
C) Organingrama
D) Tabel Grila
E) efortul