

1. Hamming pe 7 biti este un cod:
A. detector de erori
B. detector si corector de erori de 3 erori
C. de paritate incrucisata
☒ D. detector de 1 eroare, corector de 1 eroare
2. La conversia din B_{10} la B_2 , partea fractionara a numarului:
A. Se imparte la baza și se rețin resturile în ordinea LIFO
B. Se imparte la baza și se rețin resturile în ordinea FIFO
C. Se înmulțește la baza și se rețin părțile întregi în ordinea LIFO
☒ D. Se înmulțește la baza și se rețin părțile întregi în ordinea FIFO
3. La conversia din B_{10} la B_2 , partea întreagă a numărului:
☒ A. Se imparte la baza și se rețin resturile în ordinea LIFO
B. Se imparte la baza și se rețin resturile în ordinea FIFO
C. Se înmulțește la baza și se rețin părțile întregi în ordinea LIFO
D. Se înmulțește la baza și se rețin părțile întregi în ordinea FIFO
4. La conversia din B_{10} la B_2 , algoritmul pt partea fractionara a numarului se încheie:
A. niciodată;
☒ B. când se obține 0;
C. când avem atâtea cifre câte are cuvântul calculator;
☒ D. când facem conversia de 4 b pentru fiecare zecimală.
5. Cel mai mare număr pozitiv pe 8 biti este:
A. +128
B. +127
☒ C. +255
D. +256
6. Cel mai mic număr negativ pe 8 biti este:
☒ A. -128
B. -127
C. -255
D. -256
7. Se aduna bit cu bit cu bitul de semn la adunarea în VF:
A. CD
☒ B. CI
☒ C. CC
D. la toate
8. VF a fost creata pt reprezentarea numerelor:
A. naturale
☒ B. întregi positive
C. întregi negative
D. reale
9. Algebra Boole se bazeaza pe:
A. ortogonalitatea polinoamelor
B. teoria codurilor polinomiale
C. o latice distributiva doar cu prim si ultim element
☒ D. o latice distributiva cu prim si ultim element si negatie

10. Care afirmatii sunt adevarate despre FCC:
- contine termeni elementari
 - ia valoarea logica 1
 - ia valoarea logica 0
 - contine termeni cu toate variabilele functiei
11. In diagrama Karnaugh, termenii redundanti, notati cu R:
- se trec cu 0 si se iau in calcul la minimizare;
 - se trec cu 1 si se ignora;
 - se trec cu R
 - se iau in calcul la minimizare daca se obtine o suprafata mai buna, altfel se ignora.
12. Minimizarea unei FB inseamna:
- scaderea numarului de blocuri logice
 - scaderea pretului de cost al circuitului
 - cresterea numarului de blocuri logice
 - cresterea pretului de cost al circuitului
13. Rezultatul minimizării funcției de 4 variabile $f(x,y,z,t) = m_1 + m_3 + m_5 + m_9$, având termenii redundanți $R = m_6, m_7, m_{11}$, este:
- $\bar{x}t + \bar{y}t$
 - $\bar{x}t + \bar{y}t + \bar{x}yz\bar{t}$
 - $(\bar{x} + t) * (\bar{y} + t)$
 - $(\bar{x} + t) * (\bar{y} + t) * (\bar{x} + y + z + \bar{t})$
14. Rezultatul minimizării funcției de 4 variabile $f(x,y,z,t) = M_0 * M_1 * M_3 * M_9$, având termenii redundanți $R = M_7, M_{11}$, este:
- $(x + y) * (y + \bar{t})$
 - $(x + y) * (y + \bar{t}) * (x + \bar{y} + \bar{z} + \bar{t})$
 - $(xy) + (y\bar{t}) + x\bar{y}\bar{z}\bar{t}$
 - $(xy) + (y\bar{t})$
15. Care funcție de 4 variabile este reprezentată în diagrama Karnaugh următoare:

$\backslash z$	0	0	1	1
$xy \backslash t$	0	1	1	0
00	1	1	1	R
01	-	1	R	-
11	-	-	-	-
10	-	1	R	1

- $f = m_0 + m_1 + m_3 + m_5 + m_9 + m_{10}, R = m_2, m_7, m_{11}$
 - $f = m_0 + m_1 + m_3 + m_5 + m_9 + m_2 + m_7 + m_{11} + m_{10}$
 - $f = M_0 * M_1 * M_3 * M_5 * M_9 * M_{10}, R = M_2, M_7, M_{11}$
 - $f = M_0 * M_1 * M_3 * M_5 * M_9 * M_2 * M_7 * M_{11} * M_{10}$
16. Numarul $(ABC)_H$ se scrie in B_2 astfel:
- 1010 1011 1100
 - 1010 1000 1001
 - 1100 1011 1001
 - 1100 1011 1010
17. Numarul $(IBC9)_H$ se scrie in B_2 astfel:
- 0001 1011 1100 1001
 - 0001 1000 1100 1001
 - 1001 1100 1011 0001
 - 1001 1011 1100 0001

18. Numarul $(1BCD)_H$ se scrie in B_2 astfel:
- ☒ A. 0001 1011 1100 1101
 - B. 0001 1000 1100 1101
 - C. 1101 1100 1011 0001
 - D. 1101 1011 1100 0001
19. Numarul $(19CD)_H$ se scrie in B_2 astfel:
- ☒ A. 0001 1001 1100 1101
 - B. 0001 1001 1100 1001
 - C. 1101 1100 1011 0001
 - D. 1101 1100 1001 0001
20. Cum se reprezintă nr -47 în VF: CD, CI, CC pe 8b:
- ☒ A. 1010 1111; 1101 0000; 1101 0001;
 - B. 0010 1111; 1101 0000; 1101 0001;
 - C. 0010 1111; 0101 0000; 1101 0001;
 - A. D. 1010 1111; 1101 0000; 0101 0001;
21. Care este reprezentarea hexazecimală a numărului -9 din VM SP?
- ☒ A. $(C110\ 0000)_H$
 - B. $(D110\ 0000)_H$
 - C. $(C110\ 0000\ 0000\ 0000)_H$
 - A. D. $(D110\ 0000\ 0000\ 0000)_H$
22. Care este reprezentarea hexazecimală a exponentului numărului -9 din VM SP?
- A. $(82)_H$
 - B. $(82\ 00)_H$
 - C. $(81)_H$
 - ☒ D. $(81\ 00)_H$
23. Care este reprezentarea numărului 235 în BCD?
- ☒ A. 0010 0011 0101
 - B. 1110 1011
 - C. 0101 0011 0010
 - A. D. 1011 1110
24. Mesajul hexa $(1EC)_H$ se transmite după codificarea polinomial ciclică (CRC) prin polinomul de generare $G[X] = X^2 + X + 1$. Care este reprezentarea binară a mesajului transmis?
- A. 111101100 011
 - B. 111101100 100
 - C. 010 111101100
 - D. 001 111101100
25. Directiva SEGMENT se utilizeaza pentru:
- A. inchiderea unui segment
 - B. deschiderea unui segment
 - C. inchiderea programului
 - D. deschiderea programului
26. Directiva ASSUME se foloseste pentru
- A. asignarea unui singur registru de segment cu segmentul corespunzator
 - B. asignarea registrilor de segment cu segmentele corespunzatoare
 - C. inchiderea programului
 - D. deschiderea segmentului de cod

27. *Link-editarea* (editarea de legături) se realizează cu programul:

- A. *td*
- ☒ B. *link*
- C. *tasm*
- D. *debug*

28. *td prog1.exe* se folosește pentru:

- A. a lansa în execuție programul *prog1.exe*
- B. a compila programul sursă
- C. a link-edita programul sursă
- D. a vizualiza codul programului executabil

29. Pași realizării unui program în asm sunt:

- ☒ A. editare program sursă
- ☒ B. compilare program sursă
- C. compilare program obiect
- D. link-editare program obiect
- E. link-editare program sursă

30. Instrucțiunile ASM:

MOV AX, segm_date
MOV DS, AX

- A. încarcă adresa *segm_date* în AX, apoi în DS
- ☒ B. încarcă segmentul de date în AX, apoi în DS
- C. mută valoarea lui DS în AX
- D. mută segmentul de date în DS

31. Instrucțiunea

MOV AX, 15

realizează:

- A. mutarea nr 15 în AX
- ☒ B. mutarea nr $(0F)_H$ în AX
- C. o adresare imediată
- D. o adresare indirectă

32. După execuția instrucțiunii

MOV AX, 14

registru AX conține:

- A. AX: 00 14
- ☒ B. AX: 00 0E
- C. AX: 0E 00
- D. AX: 14 00

33. La adresa *CS:0000* se găsește:

- A. un șir de date în segmentul de date
- B. un vector în segmentul de stivă
- C. o instrucțiune în segmentul de cod
- D. o instrucțiune în segmentul de date

34. Instrucțiunea:

ADD AX, 3

- ☒ A. adună 3 la conținutul registrului AX,
- B. mută valoarea 3 în AX
- C. rezultatul se suprascrive peste AX
- D. AX își păstrează valoarea inițială

35. Ce face instrucțiunea?

MOV SI, OFFSET sir_date

- ☒ A. încarcă deplasarea *sir_date* față de începutul segmentului în SI
- B. încarcă *sir_date* în SI
- C. încarcă SI în *sir_date*
- D. încarcă adresa de început a lui *sir_date* în SI

36. Ce face secventa de cod?

```
.....  
MOV CX, 2  
et:  
instr_1  
LOOP et
```

- ☒ A. efectueaza *instr_1* si atata timp cat CX nu este 0
☐ B. decrementeaza automat CX
☐ C. incrementeaza automat CX
☐ D. efectueaza *instr_2* indiferent de CX

37. Ce face instructiunea *LOOP*?

- ☐ A. efectuează o ciclare
☒ B. decrementează CX cu 1 la fiecare trecere prin LOOP
☐ C. efectuează adunare
☐ D. incrementează CX cu 1 la fiecare trecere prin LOOP

38. Ce fac urmatoarele 2 instructiuni:

```
MOV AX, 4C00H  
INT 21H
```

- ☒ A. apeleaza intreruperea software 21H cu functia 4C
☐ B. asigura revenirea programului la promptul DOS
☐ C. apeleaza intreruperea hardware 21H cu functia 4C
☐ D. asigura afisarea unui caracter pe ecran

39. Care este harta memoriei pentru secventa de cod:

```
s1 SEGMENT  
a DW 15  
b DB 10  
c DW 11
```

- A. 00 15 00 10 00 11
☒ B. 00 0F 00 0A 00 0B
C. 15 00 10 11 00 00
D. 0F 00 0A 0B 00 00

40. Se dă secvența de cod:

```
s1 SEGMENT
  a DW 14
  ....
s2 SEGMENT
  .....
  MOV AX, a
  ....
  SUB AX, 2
  .....
```

Cum arata registrul AX in cele 2 cazuri?

- A. AX= 00 14; AX= 00 12
- ☒ B. AX= 00 0E; AX= 00 0C
- C. AX= 14 00; AX= 12 00
- ☒ D. AX= 0E 00; AX= 0C 00

41. Se dă secvența de cod:

```
s1 SEGMENT
  b DB 11
  ....
s2 SEGMENT
  .....
  MOV BL, b
  ....
  ADD BL, 2
  .....
```

Cum arata registrul BX in cele 2 cazuri? (nn=orice numar)

- ☒ A. BX= nn 0B; BX= nn 0D
- B. BX= nn 11; BX= nn 13
- C. BX= 0B 00; BX= 0D 00
- D. BX= 11 00; BX= 13 00

42. Ce valoare au registrii AX, CX la iesirea din LOOP?

```
.....
MOV AX, 0
MOV CX, 2
et:
  ADD AX, 2
  LOOP et
.....
```

- ☒ A. AX= 00 04; CX= 00 00
- B. AX= 00 00; CX= 00 02
- C. AX= 04 00 ; CX= 00 00
- D. AX= 04 00 ; CX= 02 00

Handwritten notes: "0", "0", "0", "B", "A", "0" (likely representing binary or hexadecimal values).

43. Știind că un microsistem lucrează în mod real, având la un moment dat CS=(F303)_H, DS=(8334)_H, IP=(6570)_H, să se calculeze *adresa fizică a operandului „data”* (când adresa efectivă este (B215)_H), precum și *adresa fizică a următoarei instrucțiuni care se va executa*.
- A. adr fiz “data”= 8E555; adr fiz urm instr= F95A0
 - B. adr fiz “data”= F95A0; adr fiz urm instr= 8E555
 - C. adr fiz “data”= BA484; adr fiz urm instr= 74A03
 - D. adr fiz “data”= 74A03; adr fiz urm instr= BA484
44. Determinați capacitatea unui hard-disk, știind că are: 40 piste, 30 fețe, 10 sectoare, 512B/sector.
- A. Aproximativ 6GB;
 - B. Aproximativ 4GB;
 - C. Aproximativ 6000MB;
 - D. Aproximativ 4000MB;