

*Timp de lucru: 2 ore.*

*Se acordă un punct din oficiu.*

1. (0.5p) Reprezentați în baza 2, pe 16 biți, numerele întregi 216 și -137.
2. (0.5p) Scrieți ce se afișează în urma executării următoarei secvențe de cod:

```
int n=3, m=4, x=1; int a[n][m], i, j;
for (i=0; i<n; i++)
{for (j=0; j<m; j++) {a[i][j] = x; x++; printf("%d ",a[i][j]);}printf("\n");}
i=2; for (j=0; j<m; j++) printf("%d ", (*(a+i)+j));
printf("\n%d %d", (*(a))[2], (*(a+1))[2]);
```

3. (0.5p) Scrieți ce se afișează în urma executării următoarei secvențe de cod:

```
int x= 10>>3 | (6 & 5), y= 3<<2;
printf ("%d %d %d %d", x, y, x && y, x ^ y);
```

4. (1p) Scrieți ce se afișează la rularea programului următor:

```
#include <stdio.h>
#define MACRODEF_1(x) x + 2*(x);
#define MACRODEF_2(x) (x + 2*x);
int main() { int x;
x=1; int y1 = MACRODEF_1(2*x); printf ("%d %d \n", y1, x);
x=2; int y2 = MACRODEF_1(x-1); printf ("%d %d \n", y2, x);
x=3; int y3 = MACRODEF_1(x++); printf ("%d %d \n", y3, x);

x=4; int y4 = MACRODEF_2(x+3); printf ("%d %d \n", y4, x);
x=5; int y5 = MACRODEF_2(x++); printf ("%d %d \n", y5, x);}
```

5. (0.75p) Scrieți un program în care definiți și apoi apelați o funcție cu număr variabil de parametri care returnează cea mai mare valoare dintr-un șir de  $n$  întregi.
6. (0.75p) Scrieți un program în care definiți și apoi apelați (pentru două exemple concrete) o funcție generică care returnează poziția într-un tablou unidimensional cu  $n$  elemente de dimensiune  $dim$  a primei apariții a unei valori  $x$  date, sau -1, dacă  $x$  nu se găsește printre elementele vectorului.



7. (1p) Pe prima linie a fișierului text *litere.txt* sunt scrise două litere separate prin spațiu, iar pe următoarea linie este scris un text cu cel mult 1000 de caractere. Scrieți un program care să citească datele de intrare, apoi să scrie textul pe ecran înlocuind în toate cuvintele litera citită de pe prima linie cu cea de-a doua literă.
8. (2.5p) Scrieți un program care gestionează date despre un grup de cel mult 30 de studenți. Pentru fiecare student se memorează numele (cel mult 50 de caractere, cuvinte separate prin spații), 3 note (numere întregi între 0 și 10) și media calculată a celor 3 note (media va fi 0 dacă cel puțin una dintre note este sub 5).  
Programul trebuie să implementeze funcții pentru fiecare dintre următoarele operații:
- a) citirea numărului de studenți și a datelor acestora (implementarea tabloului cu studenți se face dinamic, după ce se citește numărul de studenți. În plus, în tablou se alocă dinamic spațiu de memorie pentru câmpul nume);
  - b) afișarea datelor tuturor studenților;
  - c) sortarea listei de studenți în ordinea alfabetică a numelor (qsort);
  - d) sortarea listei de studenți în ordinea descrescătoare a mediilor (qsort);
  - e) afișarea în ordine alfabetică a studenților care au obținut cea mai mare medie.
9. (1.5p) Un **fișier text** conține mai multe linii de text. Orice linie din fișier conține cel mult 500 de caractere. Fiecare linie conține cuvinte separate prin separatorii uzuali. Fiecare cuvânt este format din cel mult 20 de caractere.  
Scrieți un program care:
- a) citește de la tastatură numele fișierului text,
  - b) citește datele din fișier, linie cu linie și
  - c) scrie în **fișierul binar** *linie.out* numărul de ordine al liniei care are cele mai multe cuvinte.
- Dacă sunt mai multe asemenea linii, se va scrie numărul de ordine al primei.  
Se vor trata erorile la deschiderea/ închiderea fișierelor.