

Tematica seminarului

Seminarul isi propune initierea studentilor in problematica lucrului cu masive de date (tablouri). Se vor exemplifica masivele de date unidimensionale (vectori). Se vor scrie functii pentru operatii cu elementele unui vector.

Problema 1.

Sa se scrie un program C in care se declara un vector cu maxim 20 de elemente de tip intreg. Numarul efectiv de elemente ale vectorul va fi preluat de la tastatura cu ajutorul unei functii care realizeaza si validarea acestuia: valorile admisibile sunt cuprinse in intervalul inchis $[1, 20]$. De asemenea se vor scrie alte doua functii, una pentru citirea elementelor vectorului si una pentru afisarea lor pe ecran. In programul principal se vor apela cele 3 functii.

Problema 2.

Sa se scrie un program C in care se declara un vector cu maxim 20 de elemente de tip intreg. Numarul efectiv de elemente ale vectorul va fi preluat de la tastatura cu ajutorul unei functii care realizeaza si validarea acestuia: valorile admisibile sunt cuprinse in intervalul inchis $[1, 20]$. De asemenea se vor scrie functii pentru:

- citirea elementelor vectorului de numere întregi ;
- afisarea elementelor vectorului de numere întregi ;
- calculul sumei elementelor vectorului;
- calculul produsului elementelor vectorului;
- calculul mediei aritmetice a elementelor vectorului;
- calculul numarului de elemente negative din vector;
- calculul numarului de elemente pozitive din vector;
- calculul numarului de elemente egale cu zero din vector;
- calculul valorii minime dintre elementele vectorului;
- calculul valorii maxime dintre elementele vectorului;
- calculul sumei elementelor pozitive ale vectorului;
- calculul sumei elementelor negative ale vectorului;
- cautarea unei anumite valori în vector cu returnarea pozitiei pe care se gaseste; în cazul în care valoarea se afla de mai multe ori pe pozitii diferite, vom avea ca rezultat prima pozitie pe care acesta valoare a fost gasita ; daca valoarea cautata nu este gasita in vector functia va avea ca rezultat -1;

Rezolvări.

Problema 1.

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

int citeste_nr() {
    int n;
    printf("Introduceti numarul de elemente:");
    scanf("%d", &n);
    while ( n > 20 || n < 1 ) {
        printf("Valoare eronata\n");
        printf("Introduceti numarul de elemente:");
        scanf("%d", &n);
    }
    return n;
}

void citeste(int nr, int* a) {
    int i;
    for(i = 0; i < nr; i++) {
        printf("Introduceti a[%d]=", i);
        scanf("%d", &a[i]);
    }
}

void afiseaza(int nr, int* a) {
    int i;
    for( i =0; i < nr; i++) {
        printf("a[%d]=%d\n", i , a[i]);
    }
}

int main() {
    int a[20];
    int nr_elem;

    nr_elem = citeste_nr();
    citeste(nr_elem, a);
    afiseaza(nr_elem, a);

    return 0;
}
```

Problema 2.

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

int citeste_nr() {
    int n;
    printf("Introduceti n =");
    scanf("%d", &n);
    while ( n > 20 || n < 1) {
        printf("Valoare eronata!\n");
        printf("Introduceti n =");
        scanf("%d", &n);
    }
    return n;
}

void citeste(int nr, int* a) {
    int i;
    for ( i = 0; i < nr; i++) {
        printf("Introduceti a[%d]=", i);
        scanf("%d", &a[i]);
    }
}

void afiseaza(int nr, int* a) {
    int i;
    for(i = 0; i < nr; i++) {
        printf("a[%d]=%d\n", i, a[i]);
    }
}

int suma(int nr, int* a) {
    int i;
    int s = 0;
    for(i = 0; i < nr; i++) {
        s += a[i];
    }
    return s;
}

int produs(int nr, int* a) {
    int i;
    int p = 1;
    for(i = 0; i < nr; i++) {
        p *= a[i];
    }
}
```

```

        return p;
    }

float media(int nr, int* a) {
    float m;

    m = (float)suma(nr, a) / nr;

    return m;
}

int nr_neg(int nr, int* a) {
    int i;
    int nn = 0;
    for(i = 0; i < nr; i++) {
        if(a[i] < 0)
            nn = nn + 1;
    }
    return nn;
}

int nr_poz(int nr, int* a) {
    int i;
    int np = 0;
    for(i = 0; i < nr; i++) {
        if( a[i] > 0 )
            np = np + 1;
    }
    return np;
}

int nr_zero(int nr, int* a) {
    int zero = 0;
    int i;

    for(i = 0; i < nr; i++)
        if(a[i] == 0)
            zero++;
    return zero;
}

int min(int nr, int* a) {
    int min;
    int i;
    min = a[0];
    for(i = 1; i < nr; i++)
        if( a[i] < min )

```

```

        min = a[i];
    return min;
}

int max(int nr, int* a) {
    int max;
    int i;
    max = a[0];
    for(i = 1; i < nr; i++)
        if( a[i] > max )
            max = a[i];
    return max;
}

int suma_neg(int nr, int* a)  {
    int i;
    int suma_n = 0;
    for(i = 0; i < nr; i++) {
        if( a[i] < 0 )
            suma_n += a[i];
    }
    return suma_n;
}

int suma_poz(int nr, int* a)  {
    int i;
    int suma_p = 0;
    for(i = 0; i < nr; i++) {
        if( a[i] > 0 )
            suma_p += a[i];
    }
    return suma_p;
}

int cautare(int nr, int* a, int val) {
    int i;
    for(i = 0; i < nr; i++)
        if( a[i] == val)
            return i;
    return -1;
}

int main() {
    int a[20];
    int nr_elem, sum, prod, nr_nn, nr_np, nr_nz, min_val, max_val, sum_n,
    sum_p, valoare, pos;
    float med;

```

```

nr_elem = citește_nr();
citește(nr_elem, a);
afisează(nr_elem, a);

sum = suma(nr_elem, a);
printf("Suma elementelor vectorului este = %d\n", sum);

prod = produs(nr_elem, a);
printf("Produsul elementelor vectorului este = %d\n", prod);

med = media(nr_elem, a);
printf("Media elementelor vectorului este = %f\n", med);

nr_nn = nr_neg(nr_elem, a);
printf("Numarul elementelor negative este: %d\n", nr_nn);

nr_np = nr_poz(nr_elem, a);
printf("Numarul elementelor pozitive este: %d\n", nr_np);

nr_nz = nr_zero(nr_elem, a);
printf("Numarul elementelor egale cu zero este: %d\n", nr_nz);

min_val = min(nr_elem, a);
printf("Valoarea minima este: %d\n", min_val);

max_val = max(nr_elem, a);
printf("Valoarea maxima este: %d\n", max_val);

sum_n = suma_neg(nr_elem, a);
printf("Suma elementelor negative este: %d\n", sum_n);

sum_p = suma_poz(nr_elem, a);
printf("Suma elementelor pozitive este: %d\n", sum_p);

printf("Introduceti valoarea pe care doriti sa o cautati in vector:");
scanf("%d", &valoare);
pos = cautare(nr_elem, a, valoare);
if(pos >= 0)
    printf("Valoarea %d se gaseste pe pozitia %d", valoare, pos);
else
    printf("Valoarea %d nu se gaseste in vector", valoare);

return 0;
}

```