

PROGRAMARE PROCEDURALĂ

– LABORATOR NR. 1 –

1. Fie v un tablou unidimensional format din n numere întregi. Definiți complet următoarele funcții:
 - a) *citire* – citește valoarea lui n și apoi cele n elemente ale tabloului v ;
 - b) *afișare* – afișează elementele tabloului v pe o linie, despărțite prin câte un spațiu;
 - c) *valpoz* – construiește un tablou unidimensional format din valorile strict pozitive din v ;
 - d) *semn* – schimbă semnul fiecărui element al tabloului v .

Scrieți un program care citește tabloul v și afișează pe o linie valorile strict negative din vectorul v și pe o altă linie valorile strict pozitive din tabloul v , folosind apeluri utile ale funcțiilor definite anterior. **Toate tablourile vor fi alocate dinamic!**

Rezolvare:

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

void afisare(int *v, int n)
{
    int i;

    for (i = 0; i < n; i++)
        printf("%d ", v[i]);
    printf("\n");
}

void semn(int *v, int n)
{
    int i;

    for(i = 0; i < n; i++)
        v[i] = -v[i];
}

void citire(int **v, int *n)
{
    int i;
    printf("n = ");
    scanf("%d", n);
```

```

    *v = malloc(*n*sizeof(int));

    printf("Introduceti elementele lui v: \n");
    for(i = 0; i < *n; i++)
        scanf("%d", &(*v)[i]);
}

void valpoz(int *v, int n, int **vpoz, int *nrpoz)
{
    int i;

    *nrpoz = 0;
    for (i = 0; i < n; i++)
        if(v[i] > 0) (*nrpoz)++;

    *vpoz = malloc(*nrpoz * sizeof(int));

    *nrpoz = 0;
    for (i = 0; i < n; i++)
        if(v[i] > 0)
        {
            (*vpoz)[*nrpoz] = v[i];
            (*nrpoz)++;
        }
}

int main()
{
    int n, nrpoz;
    int *v, *vpoz;

    citire(&v, &n);
    printf("Afisare vector v: \n");

    afisare(v, n);
    valpoz(v, n, &vpoz, &nrpoz);

    printf("Afisare vector vpoz: \n");
    afisare (vpoz, nrpoz);

    free(v);
    free(vpoz);

    return 0;
}

```