Ministerul Educaţiei și Cercetăriial Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Informatică și Ingineria a sistemelor

Disciplina: Programarea Calculatoarelor

RAPORT

Lucru Individual

A efectuat : **Veaceslav Mihalachi**

#Probleme pentru tablouri unidimensionale:

Input

```c
#include <stdio.h>

int main() {
    int t[3][3];
    int n = 3, m = 3;
    int count[3] = {0};

    printf("introduceti numere pentru matricea 3x3:\n\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            scanf("%d", &t[i][j]);}}

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            if(t[i][j] > 0) {
                count[i]++;}} }

    for(int i = 0; i < n-1; i++) {
        for(int j = i+1; j < n; j++) {
            if(count[i] > count[j]) {
                int temp_c = count[i];
                count[i] = count[j];
                count[j] = temp_c;

                for(int k = 0; k < m; k++) {
                    int temp = t[i][k];
                    t[i][k] = t[j][k];
                    t[j][k] = temp;}}}}
    printf("tabloul ordonat:\n\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            printf("%d ", t[i][j]);}
        printf("\n");}
    return 0;}

```

Output

```
introduceti numere pentru matricea 3x3:

-1 2 5
9 5 5
-4 -7 3
tabloul ordonat:

-4 -7 3
-1 2 5
9 5 5
```

Input

main.c

```c
#include <stdio.h>

void citesteTablou(int arr[], int n, int i) {
    if(i < n) {
        printf("Element %d: ", i+1);
        scanf("%d", &arr[i]);
        citesteTablou(arr, n, i+1);
    }
}

void afiseazaTablou(int arr[], int n, int i) {
    if(i < n) {
        printf("%d ", arr[i]);
        afiseazaTablou(arr, n, i+1);
    }
}

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    citesteTablou(arr, n, 0);

    printf("Tabloul: ");
    afiseazaTablou(arr, n, 0);

    return 0;
}
```

Output

```
Numar elemente: 3
Element 1: 2
Element 2: 4
Element 3: 1
Tabloul: 2 4 1
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Elemente negative: ");
    for(int i = 0; i < n; i++) {
        if(arr[i] < 0) {
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```

Output

```
Numar elemente: 3
Introduceti 3 elemente:
-4
2
1
Elemente negative: -4
```

Input

main.c

```c
#include <stdio.h>

int sumaRecursiv(int arr[], int n) {
    if(n <= 0) return 0;
    return arr[n-1] + sumaRecursiv(arr, n-1);
}

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Suma: %d\n", sumaRecursiv(arr, n));

    return 0;
}
```

Output

```
Numar elemente: 4
Introduceti 4 elemente:
13
-2
3
7
Suma: 21
```

Input

main.c

```c
#include <stdio.h>
#include <limits.h>

int maximRecursiv(int arr[], int n) {
    if(n == 1) return arr[0];

    int max_rest = maximRecursiv(arr, n-1);
    return (arr[n-1] > max_rest) ? arr[n-1] : max_rest;
}

int minimRecursiv(int arr[], int n) {
    if(n == 1) return arr[0];

    int min_rest = minimRecursiv(arr, n-1);
    return (arr[n-1] < min_rest) ? arr[n-1] : min_rest;
}

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Maxim: %d\n", maximRecursiv(arr, n));
    printf("Minim: %d\n", minimRecursiv(arr, n));

    return 0;
}
```

Output

```
Numar elemente: 3
Introduceti 3 elemente:
2
5
3
Maxim: 5
Minim: 2
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int pare = 0, impare = 0;
    for(int i = 0; i < n; i++) {
        if(arr[i] % 2 == 0) pare++;
        else impare++;
    }

    printf("Pare: %d\nImpare: %d\n", pare, impare);

    return 0;
}
```

Output

```
Numar elemente: 3
Introduceti 3 elemente:
3
7
10
Pare: 1
Impare: 2
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int negative = 0;
    for(int i = 0; i < n; i++) {
        if(arr[i] < 0) negative++;
    }

    printf("Elemente negative: %d\n", negative);

    return 0;
}
```

Output

```
Introduceti 5 elemente:
-4
-2
1
5
-8
Elemente negative: 3
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int sursa[n];
    int destinatie[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &sursa[i]);
    }

    for(int i = 0; i < n; i++) {
        destinatie[i] = sursa[i];
    }

    printf("Tabloul copiat: ");
    for(int i = 0; i < n; i++) {
        printf("%d ", destinatie[i]);
    }

    return 0;
}
```

Output

```
Numar elemente: 3
Introduceti 3 elemente:
4
-1
87
Tabloul copiat: 4 -1 87
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente initiale: ");
    scanf("%d", &n);

    int arr[n+1];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int valoare, pozitie;
    printf("Valoare de inserat: ");
    scanf("%d", &valoare);
    printf("Pozitie (0-%d): ", n);
    scanf("%d", &pozitie);
    for(int i = n; i > pozitie; i--) {
        arr[i] = arr[i-1];
    }
    arr[pozitie] = valoare;

    printf("Tabloul dupa inserare: ");
    for(int i = 0; i <= n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output

```
Numar elemente initiale: 4
Introduceti 4 elemente:
20
10
4
3
Valoare de inserat: 2
Pozitie (0-4): 1
Tabloul dupa inserare: 20 2 10 4 3
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int pozitie;
    printf("Pozitie de sters (0-%d): ", n-1);
    scanf("%d", &pozitie);
        for(int i = pozitie; i < n-1; i++) {
        arr[i] = arr[i+1];
    }

    printf("Tabloul dupa stergere: ");
    for(int i = 0; i < n-1; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output

```
Numar elemente: 3
Introduceti 3 elemente:
13
-25
90
Pozitie de sters (0-2): 2
Tabloul dupa stergere: 13 -25
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Frecventa elementelor:\n");

    for(int i = 0; i < n; i++) {
        if(arr[i] == -1) continue;

        int count = 1;
        for(int j = i+1; j < n; j++) {
            if(arr[i] == arr[j]) {
                count++;
                arr[j] = -1;
            }
        }
        printf("%d apare de %d ori\n", arr[i], count);
    }
    return 0;
}
```

Output

```
11
3
11
24
Frecventa elementelor:
11 apare de 2 ori
3 apare de 1 ori
24 apare de 1 ori
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Elemente unice: ");

    for(int i = 0; i < n; i++) {
        int esteUnic = 1;
        for(int j = 0; j < n; j++) {
            if(i != j && arr[i] == arr[j]) {
                esteUnic = 0;
                break;
            }
        }
        if(esteUnic) {
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```

Output

```
Numar elemente: 4
Introduceti 4 elemente:
13
13
2
-4
Elemente unice: 2 -4
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int duplicate = 0;

    for(int i = 0; i < n; i++) {
        if(arr[i] == -1) continue;

        int count = 1;
        for(int j = i+1; j < n; j++) {
            if(arr[i] == arr[j]) {
                count++;
                arr[j] = -1;
            }
        }
        if(count > 1) {
            duplicate += (count - 1);
        }
    }

    printf("Numar total de duplicate: %d\n", duplicate);

    return 0;
}
```

Output

```
Introduceti 3 elemente:
25
25
1
Numar total de duplicate: 1


...Program finished with exit code 0
Press ENTER to exit console.
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    for(int i = 0; i < n; i++) {
        for(int j = i+1; j < n; j++) {
            if(arr[i] == arr[j]) {
                for(int k = j; k < n-1; k++) {
                    arr[k] = arr[k+1];
                }
                n--;
                j--;
            }
        }
    }

    printf("Tablou fara duplicate: ");
    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output

```
Numar elemente: 4
Introduceti 4 elemente:
13
-2
24
13
Tablou fara duplicate: 13 -2 24
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n1, n2;

    printf("Dimensiune primul tablou: ");
    scanf("%d", &n1);
    int arr1[n1];
    printf("Introduceti %d elemente:\n", n1);
    for(int i = 0; i < n1; i++) {
        scanf("%d", &arr1[i]);
    }

    printf("Dimensiune al doilea tablou: ");
    scanf("%d", &n2);
    int arr2[n2];
    printf("Introduceti %d elemente:\n", n2);
    for(int i = 0; i < n2; i++) {
        scanf("%d", &arr2[i]);
    }

    int arr3[n1 + n2];
        for(int i = 0; i < n1; i++) {
        arr3[i] = arr1[i];
    }
        for(int i = 0; i < n2; i++) {
        arr3[n1 + i] = arr2[i];
    }

    printf("Tabloul imbunatit: ");
    for(int i = 0; i < n1 + n2; i++) {
        printf("%d ", arr3[i]);
    }

    return 0;
}
```

Output

```
13
-2
24
13
Tablou fara duplicate: 13 -2 24
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
        for(int i = 0; i < n/2; i++) {
        int temp = arr[i];
        arr[i] = arr[n-i-1];
        arr[n-i-1] = temp;
    }

    printf("Tablou inversat: ");
    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output

```
Introduceti 3 elemente:
13
19
2
Tablou inversat: 2 19 13
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int pare[n], impare[n];
    int count_pare = 0, count_impare = 0;

    for(int i = 0; i < n; i++) {
        if(arr[i] % 2 == 0) {
            pare[count_pare++] = arr[i];
        } else {
            impare[count_impare++] = arr[i];
        }
    }
    printf("Elemente pare: ");
    for(int i = 0; i < count_pare; i++) {
        printf("%d ", pare[i]);
    }
    printf("\nElemente impare: ");
    for(int i = 0; i < count_impare; i++) {
        printf("%d ", impare[i]);
    }

    return 0;
}
```

Output

```
Introduceti 3 elemente:
3
15
10
Elemente pare: 10
Elemente impare: 3 15
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);

    int arr[n];

    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int element;
    printf("Element de cautat: ");
    scanf("%d", &element);

    printf("Pozitii gasite: ");
    int gasit = 0;

    for(int i = 0; i < n; i++) {
        if(arr[i] == element) {
            printf("%d ", i);
            gasit = 1;
        }
    }

    if(!gasit) {
        printf("Elementul nu a fost gasit!");
    }

    return 0;
}
```

Output

```
Introduceti 4 elemente:
14
18
21
28
Element de cautat: 21
Pozitii gasite: 2
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Numar elemente: ");
    scanf("%d", &n);
    int arr[n];
    printf("Introduceti %d elemente:\n", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int alegere;
    printf("1 - Crescator\n2 - Descrescator\nAlegere: ");
    scanf("%d", &alegere);
        for(int i = 0; i < n-1; i++) {
        for(int j = 0; j < n-i-1; j++) {
            if((alegere == 1 && arr[j] > arr[j+1]) ||
               (alegere == 2 && arr[j] < arr[j+1])) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    printf("Tablou sortat: ");
    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output

```
-2
24
37
11
9
1 - Crescator
2 - Descrescator
Alegere: 1
Tablou sortat: -2 9 11 24 37
```

#Probleme pentru tablouri bidimensionale:

main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int** alocaMatrice(int n, int m) {
    int **matrix = (int **)malloc(n * sizeof(int *));
    if (matrix == NULL) {
        printf("Eroare la alocarea memoriei pentru linii.\n");
        return NULL;}
    for (int i = 0; i < n; i++) {
        matrix[i] = (int *)malloc(m * sizeof(int));
        if (matrix[i] == NULL) {
            printf("Eroare la alocarea memoriei pentru coloane.\n");
            for (int j = 0; j < i; j++) {
                free(matrix[j]); }
            free(matrix);
            return NULL;}}
    return matrix;}

void elibereazaMatrice(int **matrix, int n) {
    if (matrix != NULL) {
        for (int i = 0; i < n; i++) {
            free(matrix[i]);}
        free(matrix);}}

void citesteMatrice(int **matrix, int n, int m) {
    printf("Introduceți elementele matricei (%d linii x %d coloane):\n", n, m);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) { printf("", i, j);
            scanf("%d", &matrix[i][j]);}}}

void completeazaAleator(int **matrix, int n, int m) {
    srand(time(NULL));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            matrix[i][j] = rand() % 100;}}
    printf("Matricea a fost completată cu valori aleatorii!\n");}

void insertionSort(int v[], int size) {
    for (int i = 1; i < size;i++) {
        int key = v[i];
        int j = i - 1;
        while (j >= 0 && v[j] >key) {
            v[j + 1] = v[j]; j--;
        }
        v[j+ 1] = key;}}
void sorteazaPerimetru(int **matrix, int n, int m) {
    int size = 2*(n + m) - 4;
    int *v = (int*)malloc(size * sizeof(int));
    int k = 0;
    for (int j = 0; j < m; j++) v[k++] = matrix[0][j];
    for (int i = 1; i < n-1; i++) v[k++] = matrix[i][m-1];
    for (int j = m-1; j >= 0; j--) v[k++] = matrix[n-1][j];
    for (int i = n-2; i > 0; i--) v[k++] = matrix[i][0];
    insertionSort(v, size);
```

main.c

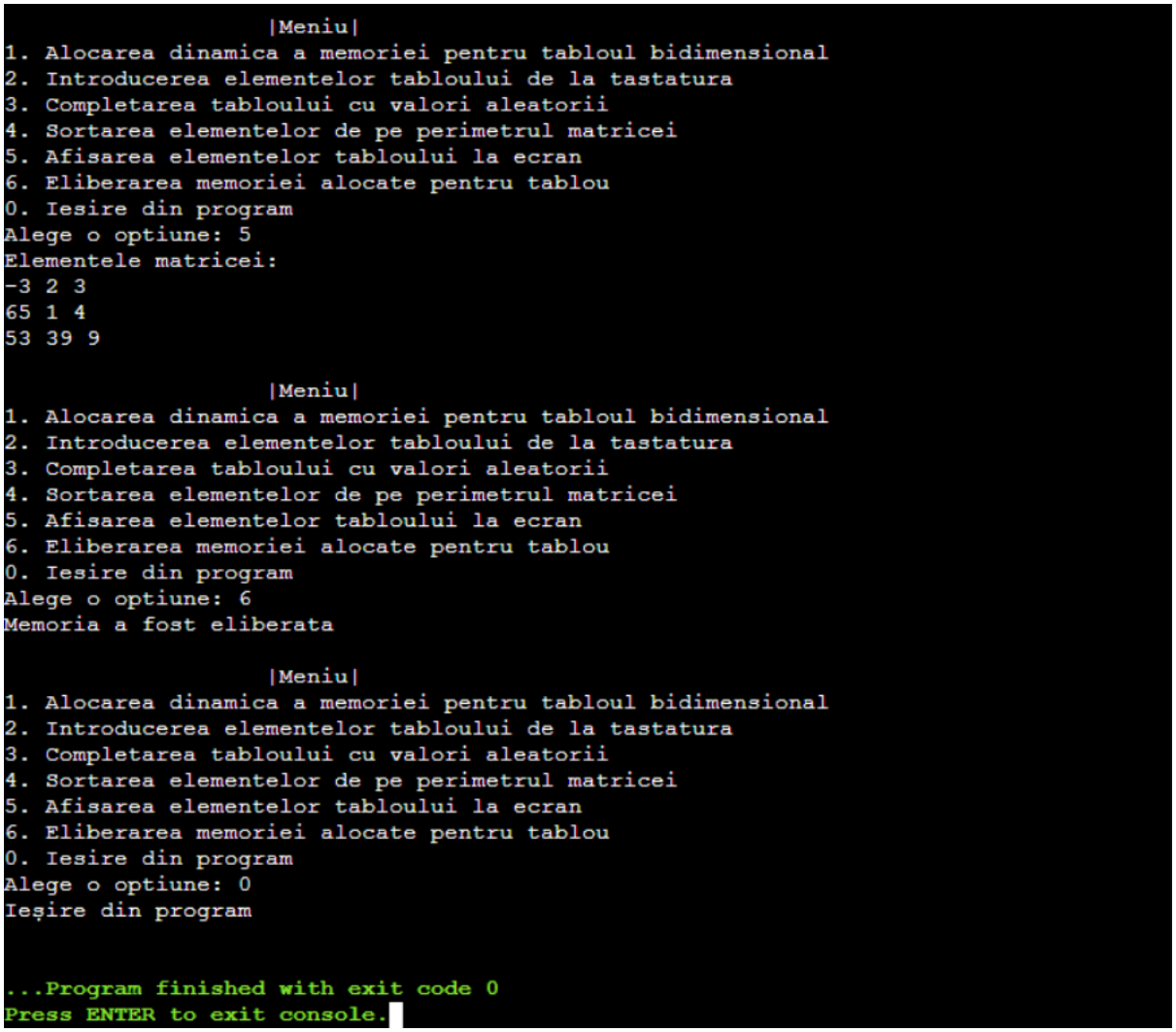
```c
        v[j+ 1] = key;}}
void sorteazaPerimetru(int **matrix, int n, int m) {
    int size = 2*(n + m) - 4;
    int *v = (int*)malloc(size * sizeof(int));
    int k = 0;
    for (int j = 0; j < m; j++) v[k++] = matrix[0][j];
    for (int i = 1; i < n-1; i++) v[k++] = matrix[i][m-1];
    for (int j = m-1; j >= 0; j--) v[k++] = matrix[n-1][j];
    for (int i = n-2; i > 0; i--) v[k++] = matrix[i][0];
    insertionSort(v, size);
    k = 0;
    for (int j = 0; j < m; j++) matrix[0][j] = v[k++];
    for (int i = 1; i < n-1; i++) matrix[i][m-1] = v[k++];
    for (int j = m-1; j >= 0; j--) matrix[n-1][j] = v[k++];
    for (int i = n-2; i > 0; i--) matrix[i][0] = v[k++];
    free(v);
    printf("Perimetrul matricei a fost sortat ascendent.\n");}

void afiseazaMatrice(int **matrix, int n, int m) {
    if (matrix == NULL) {
        printf("Matricea nu este alocată.\n");
        return;  }

    printf("Elementele matricei:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            printf("%d ", matrix[i][j]);}
        printf("\n");}}

int main() {
    int n = 0, m = 0;
    int **matrix = NULL;
    int optiune;

    do {
        printf("\n                    |Meniu| \n");
        printf("1. Alocarea dinamica a memoriei pentru tabloul bidimensional\n");
        printf("2. Introducerea elementelor tabloului de la tastatura\n");
        printf("3. Completarea tabloului cu valori aleatorii \n");
        printf("4. Sortarea elementelor de pe perimetrul matricei\n");
        printf("5. Afisarea elementelor tabloului la ecran\n");
        printf("6. Eliberarea memoriei alocate pentru tablou\n");
        printf("0. Iesire din program\n");printf("Alege o optiune: ") ;
        scanf("%d",&optiune);
```

main.c

```c
        switch (optiune){
            case 1:
                if (matrix != NULL) {
                    printf("Matricea este deja alocată. Elibereaz-o mai întai.\n");}else {
                    printf("Introduceți numărul de linii : ");
                    scanf("%d", &n);
                    printf("Introduceți numărul de coloane : ");
                    scanf("%d", &m);
                    matrix = alocaMatrice(n, m);
                    if (matrix!= NULL) {
                        printf("Memoria a fost alocată cu succes.\n");}} break;
            case 2:
                if (matrix == NULL) {
                    printf("Matricea nu e alocată. Alege opțiunea 1 mai întai.\n");} else {
                    citesteMatrice(matrix, n, m);}
                break;
            case 3:
                if (matrix ==NULL) {
                    printf("Matricea nu este alocată. Alege opțiunea 1 mai întai.\n");} else {
                    completeazaAleator(matrix, n, m);}
                break;
             case 4:
                if (matrix == NULL) {
                    printf("Matricea nu este alocată. Alege opțiunea 1 mai întai.\n");} else {
                    sorteazaPerimetru(matrix, n, m);}
                break;
            case 5:
                afiseazaMatrice(matrix,n,m);
                break;
            case 6:
                elibereazaMatrice(matrix,n);
                matrix = NULL;n = 0;m =0;
                printf("Memoria a fost eliberata\n");
                break;
            case 0:
                if (matrix != NULL) {
                    elibereazaMatrice(matrix, n);
                }
                printf("Ieșire din program\n");
                break;
            default:
                printf("Opțiune invalidă. Încearca din nou!!\n");
                break;}
    } while (optiune != 0);
```

Output

```
                    |Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 1
Introduceți numărul de linii : 3
Introduceți numărul de coloane : 3
Memoria a fost alocată cu succes.

                    |Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 2
Introduceți elementele matricei (3 linii x 3 coloane):
3 65 39
4 1 -3
53 9 2

                    |Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 4
Perimetrul matricei a fost sortat ascendent.

                    |Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 5
Elementele matricei:
-3 2 3
65 1 4
53 39 9
```


|Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 5
Elementele matricei:
-3 2 3
65 1 4
53 39 9
|Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 6
Memoria a fost eliberata
|Meniu|
1. Alocarea dinamica a memoriei pentru tabloul bidimensional
2. Introducerea elementelor tabloului de la tastatura
3. Completarea tabloului cu valori aleatorii
4. Sortarea elementelor de pe perimetrul matricei
5. Afisarea elementelor tabloului la ecran
6. Eliberarea memoriei alocate pentru tablou
0. Iesire din program
Alege o optiune: 0
Ieșire din program
...Program finished with exit code 0
Press ENTER to exit console.

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int A[r][c], B[r][c], suma[r][c];
    printf("Introduceti matricea A:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Introduceti matricea B:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &B[i][j]);
        }
    }
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            suma[i][j] = A[i][j] + B[i][j];
        }
    }
    printf("Matricea suma:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", suma[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea A:
3 12
34 3
Introduceti matricea B:
12 2
34 1
Matricea suma:
15 14
68 4
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int A[r][c], B[r][c], diferenta[r][c];
    printf("Introduceti matricea A:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Introduceti matricea B:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &B[i][j]);
        }
    }
        for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            diferenta[i][j] = A[i][j] - B[i][j];
        }
    }
    printf("Matricea diferenta (A - B):\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", diferenta[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea A:
13 29
-43 3
Introduceti matricea B:
-11 -24
16 19
Matricea diferenta (A - B):
24 53
-59 -16
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int r, c, scalar;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int A[r][c], rezultat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Introduceti scalarul: ");
    scanf("%d", &scalar);
        for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            rezultat[i][j] = A[i][j] * scalar;
        }
    }
    printf("Matricea dupa multiplicare cu %d:\n", scalar);
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", rezultat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
34 1
24 5
Introduceti scalarul: 5
Matricea dupa multiplicare cu 5:
170 5
120 25
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r1, c1, r2, c2;
    printf("Matricea A:\n");
    printf("Randuri: ");
    scanf("%d", &r1);
    printf("Coloane: ");
    scanf("%d", &c1);
    printf("Matricea B:\n");
    printf("Randuri: ");
    scanf("%d", &r2);
    printf("Coloane: ");
    scanf("%d", &c2);
    if(c1 != r2) {
        printf("Nu se pot multiplica! (c1 != r2)\n");
        return 0;}
    int A[r1][c1], B[r2][c2], C[r1][c2];
    printf("Introduceti matricea A:\n");
    for(int i = 0; i < r1; i++) {
        for(int j = 0; j < c1; j++) {
            scanf("%d", &A[i][j]); }
    }
    printf("Introduceti matricea B:\n");
    for(int i = 0; i < r2; i++) {
        for(int j = 0; j < c2; j++) {
            scanf("%d", &B[i][j]); }}
        for(int i = 0; i < r1; i++) {
        for(int j = 0; j < c2; j++) {
            C[i][j] = 0;}}
        for(int i = 0; i < r1; i++) {
        for(int j = 0; j < c2; j++) {
            for(int k = 0; k < c1; k++) {
                C[i][j] += A[i][k] * B[k][j];}}}
    printf("Matricea rezultat:\n");
    for(int i = 0; i < r1; i++) {
        for(int j = 0; j < c2; j++) {
            printf("%d ", C[i][j]);}
        printf("\n");}
    return 0;
}
```

Output

```
Matricea A:
Randuri: 2
Coloane: 2
Matricea B:
Randuri: 2
Coloane: 2
Introduceti matricea A:
-13 -5
2 31
Introduceti matricea B:
2 5
7 3
Matricea rezultat:
-61 -80
221 103
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int A[r][c], B[r][c];
    printf("Introduceti matricea A:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &A[i][j]);
        }
    }
    printf("Introduceti matricea B:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &B[i][j]);
        }
    }
    int egale = 1;
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            if(A[i][j] != B[i][j]) {
                egale = 0;
                break;
            }
        }
        if(!egale) break;
    }
    if(egale) {
        printf("Matricele sunt egale.\n");
    } else {
        printf("Matricele NU sunt egale.\n");
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea A:
2 3
3 2
Introduceti matricea B:
2 3
3 2
Matricele sunt egale.
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica (n x n): ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int suma = 0;
    for(int i = 0; i < n; i++) {
        suma += mat[i][i];
    }

    printf("Suma diagonalei principale: %d\n", suma);

    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
2 4
3 1
Suma diagonalei principale: 3
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica: ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int suma = 0;
    for(int i = 0; i < n; i++) {
        suma += mat[i][n-1-i];
    }

    printf("Suma diagonalei secundare: %d\n", suma);

    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
3 5
1 7
Suma diagonalei secundare: 6
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
        printf("Suma randuri:\n");
    for(int i = 0; i < r; i++) {
        int suma_rand = 0;
        for(int j = 0; j < c; j++) {
            suma_rand += mat[i][j];
        }
        printf("Rand %d: %d\n", i+1, suma_rand);
    }
        printf("\nSuma coloane:\n");
    for(int j = 0; j < c; j++) {
        int suma_coloana = 0;
        for(int i = 0; i < r; i++) {
            suma_coloana += mat[i][j];
        }
        printf("Coloana %d: %d\n", j+1, suma_coloana);
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
23 25
51 2
Suma randuri:
Rand 1: 48
Rand 2: 53

Suma coloane:
Coloana 1: 74
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int n;
    printf("Dimensiune matrice patratica (n x n): ");
    scanf("%d", &n);
    int mat[n][n];
    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
        for(int i = 0; i < n; i++) {
        int temp = mat[i][i];
        mat[i][i] = mat[i][n-1-i];
        mat[i][n-1-i] = temp;
    }
    printf("Matricea dupa schimbarea diagonalelor:\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
2 5
6 1
Matricea dupa schimbarea diagonalelor:
5 2
1 6
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int n;
    printf("Dimensiune matrice patratica): ");
    scanf("%d", &n);
    int mat[n][n];
    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
    printf("Matricea triunghiulara superioara:\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(i <= j) {
                printf("%d ", mat[i][j]);
            } else {
                printf("0 ");
            }
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Dimensiune matrice patratica): 2
Introduceti matricea 2x2:
42 1
54 6
Matricea triunghiulara superioara:
42 1
0 6
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica: ");
    scanf("%d", &n);
    int mat[n][n];
    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
    printf("Matricea triunghiulara inferioara:\n");
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(i >= j) {
                printf("%d ", mat[i][j]);
            } else {
                printf("0 ");
            }
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
23 7
1 23
Matricea triunghiulara inferioara:
23 0
1 23
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica: ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int suma = 0;
    for(int i = 0; i < n; i++) {
        for(int j = i; j < n; j++) {
            suma += mat[i][j];
        }
    }

    printf("Suma triunghiulara superioara: %d\n", suma);

    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
1 2
3 4
Suma triunghiulara superioara: 7
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica : ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int suma = 0;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j <= i; j++) {
            suma += mat[i][j];
        }
    }

    printf("Suma triunghiulara inferioara: %d\n", suma);

    return 0;
}
```

Output

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica : ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int suma = 0;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j <= i; j++) {
            suma += mat[i][j];
        }
    }

    printf("Suma triunghiulara inferioara: %d\n", suma);

    return 0;
}
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c], transpusa[c][r];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            transpusa[j][i] = mat[i][j];
        }
    }
    printf("Matricea transpusa:\n");
    for(int i = 0; i < c; i++) {
        for(int j = 0; j < r; j++) {
            printf("%d ", transpusa[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
2 4
1 55
Matricea transpusa:
2 1
4 55
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice: ");
    scanf("%d", &n);

    if(n != 2 && n != 3) {
        printf("\n");
        return 0;
    }

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int det;
    if(n == 2) {
        det = mat[0][0] * mat[1][1] - mat[0][1] * mat[1][0];
    } else {
        det = mat[0][0]*mat[1][1]*mat[2][2] +
              mat[0][1]*mat[1][2]*mat[2][0] +
              mat[0][2]*mat[1][0]*mat[2][1] -
              mat[0][2]*mat[1][1]*mat[2][0] -
              mat[0][0]*mat[1][2]*mat[2][1] -
              mat[0][1]*mat[1][0]*mat[2][2];
    }

    printf("Determinantul: %d\n", det);

    return 0;
}
```

Output

```
1 7
Determinantul: 16
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica : ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int esteUnitar = 1;

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(i == j) {
                if(mat[i][j] != 1) esteUnitar = 0;
            } else {
                if(mat[i][j] != 0) esteUnitar = 0;
            }
        }
    }

    if(esteUnitar) {
        printf("Matricea este unitara (matrice identitate).\n");
    } else {
        printf("Matricea NU este unitara.\n");
    }

    return 0;
}
```

Output

```
Dimensiune matrice patratica : 2
Introduceti matricea 2x2:
3 5
1 7
Matricea NU este unitara.
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int zerouri = 0;
    int total = r * c;
        for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            if(mat[i][j] == 0) {
                zerouri++;
            }
        }
    }

    float procent = (float)zerouri / total * 100;
    printf("Zerouri: %d/%d (%.2f%%)\n", zerouri, total, procent);

    if(procent >= 70) {
        printf("Matricea este rara (sparse).\n");
    } else {
        printf("Matricea NU este rara.\n");
    }

    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
0 65
0 43
Zerouri: 2/4 (50.00%)
Matricea NU este rara.
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int n;
    printf("Dimensiune matrice patratica: ");
    scanf("%d", &n);

    int mat[n][n];

    printf("Introduceti matricea %dx%d:\n", n, n);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int simetrica = 1;
        for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            if(mat[i][j] != mat[j][i]) {
                simetrica = 0;
                break;
            }
        }
        if(!simetrica) break;
    }
    if(simetrica) {
        printf("Matricea este simetrica.\n");
    } else {
        printf("Matricea NU este simetrica.\n");
    }
    return 0;
}
```

Output

```
Dimensiune matrice patratica: 2
Introduceti matricea 2x2:
32 54
22 12
Matricea NU este simetrica.
```

Input

main.c

```c
#include <stdio.h>

int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);

    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
        for(int i = 0; i < r; i++) {
        int temp = mat[i][0];
        for(int j = 0; j < c-1; j++) {
            mat[i][j] = mat[i][j+1];
        }
        mat[i][c-1] = temp;
    }
    printf("Matricea dupa ROL:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
22 44
31 1
Matricea dupa ROL:
44 22
1 31
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
        for(int i = 0; i < r; i++) {
        int temp = mat[i][c-1];
        for(int j = c-1; j > 0; j--) {
            mat[i][j] = mat[i][j-1];
        }
        mat[i][0] = temp;
    }
    printf("Matricea dupa ROR:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
13 15
23 22
Matricea dupa ROR:
15 13
22 23
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
        int temp[c];
    for(int j = 0; j < c; j++) {
        temp[j] = mat[0][j];
    }
        for(int i = 0; i < r-1; i++) {
        for(int j = 0; j < c; j++) {
            mat[i][j] = mat[i+1][j];
        }
    }
        for(int j = 0; j < c; j++) {
        mat[r-1][j] = temp[j];
    }
    printf("Matricea dupa rotire sus:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

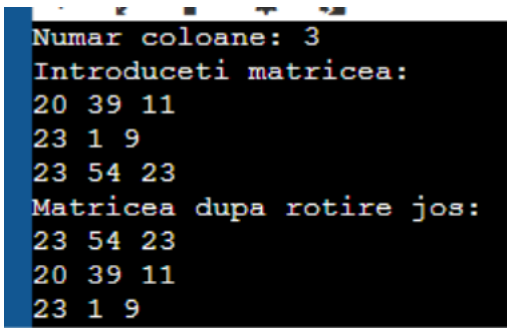
```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
24 26
35 12
Matricea dupa rotire sus:
35 12
24 26
```

Input

main.c

```c
#include <stdio.h>
int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }

    int temp[c];
    for(int j = 0; j < c; j++) {
        temp[j] = mat[r-1][j];
    }
        for(int i = r-1; i > 0; i--) {
        for(int j = 0; j < c; j++) {
            mat[i][j] = mat[i-1][j];
        }
    }
        for(int j = 0; j < c; j++) {
        mat[0][j] = temp[j];
    }

    printf("Matricea dupa rotire jos:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>

int main() {
    int r, c;
    printf("Numar randuri: ");
    scanf("%d", &r);
    printf("Numar coloane: ");
    scanf("%d", &c);
    int mat[r][c];
    printf("Introduceti matricea:\n");
    for(int i = 0; i < r; i++) {
        for(int j = 0; j < c; j++) {
            scanf("%d", &mat[i][j]);
        }
    }
    int pool_size;
    printf("Dimensiune fereastra pooling : ");
    scanf("%d", &pool_size);
        int rez_r = r / pool_size;
    int rez_c = c / pool_size;
    if(rez_r == 0 || rez_c == 0) {
        printf("Fereastra prea mare pentru aceasta matrice!\n");
        return 0;
    }

    int rezultat[rez_r][rez_c];
        for(int i = 0; i < rez_r; i++) {
        for(int j = 0; j < rez_c; j++) {
            int max_val = mat[i*pool_size][j*pool_size];
                        for(int x = 0; x < pool_size; x++) {
                for(int y = 0; y < pool_size; y++) {
                    int val = mat[i*pool_size + x][j*pool_size + y];
                    if(val > max_val) {
                        max_val = val;}}}
            rezultat[i][j] = max_val;}}
    printf("Matricea dupa max-pooling (%dx%d):\n", pool_size, pool_size);
    for(int i = 0; i < rez_r; i++) {
        for(int j = 0; j < rez_c; j++) {
            printf("%d ", rezultat[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
Numar randuri: 2
Numar coloane: 2
Introduceti matricea:
13 15
17 19
Dimensiune fereastra pooling : 2
Matricea dupa max-pooling (2x2):
19
```

#Probleme pentru șiruri de caractere:

Input

main.c File (Ctrl+M)

```c
#include <stdio.h>
#include <string.h>
#define MAX 1005
int main() {
    char str[MAX];
    char paranteze[MAX];
    int s = -1;
    int adv = 1;
    printf("Introdu șirul: ");
    fgets(str, 1000, stdin);
    str[strcspn(str, "\n")] = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == '(' || str[i] == '{' || str[i] == '[') {
            paranteze[++s] = str[i];
        } else if (str[i] == ')'|| str[i] == '}' || str[i] == ']') {
            if (s == -1){
                adv = 0;
                break;
            }
            if ((str[i] == ')' && paranteze[s] !='(')|| (str[i] == '}' && paranteze[s] !='{')||
            (str[i] == ']' && paranteze[s] !='[')){
                adv = 0;
                break;
            }
            s--;}}
    if (s != -1) {
        adv = 0;}
    if (adv) {
        printf("Parantezele sunt corecte.\n");} else {
        printf("Parantezele nu sunt corecte.\n");}

    return 0;
}
```

Output

```
Introdu șirul: {()}[]
Parantezele sunt corecte.


...Program finished with exit code 0
Press ENTER to exit console.
```

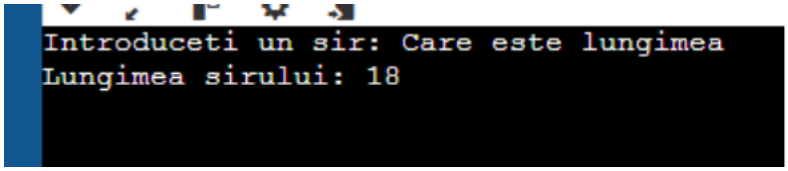
```
Introdu șirul: {0)[]
Parantezele nu sunt corecte.


...Program finished with exit code 0
Press ENTER to exit console.
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = 0;
    while(sir[lungime] != '\0' && sir[lungime] != '\n') {
        lungime++;
    }

    printf("Lungimea sirului: %d\n", lungime);

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>

int main() {
    char sir1[100], sir2[100];

    printf("Introduceti un sir: ");
    fgets(sir1, sizeof(sir1), stdin);

    int i = 0;
    while(sir1[i] != '\0' && sir1[i] != '\n') {
        i++;
    }
    sir1[i] = '\0';

    i = 0;
    while(sir1[i] != '\0') {
        sir2[i] = sir1[i];
        i++;
    }
    sir2[i] = '\0';

    printf("Sirul original: %s\n", sir1);
    printf("Sirul copiat: %s\n", sir2);

    return 0;
}
```

Output

```
Introduceti un sir: Bine ai venit
Sirul original: Bine ai venit
Sirul copiat: Bine ai venit
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir1[100], sir2[100], rezultat[200];

    printf("Introduceti primul sir: ");
    fgets(sir1, sizeof(sir1), stdin);

    printf("Introduceti al doilea sir: ");
    fgets(sir2, sizeof(sir2), stdin);
        int i = 0;
    while(sir1[i] != '\0' && sir1[i] != '\n') {
        i++;
    }
    sir1[i] = '\0';

    i = 0;
    while(sir2[i] != '\0' && sir2[i] != '\n') {
        i++;
    }
    sir2[i] = '\0';
        int j = 0;
    i = 0;
    while(sir1[i] != '\0') {
        rezultat[j++] = sir1[i++];
    }

    i = 0;
    while(sir2[i] != '\0') {
        rezultat[j++] = sir2[i++];
    }
    rezultat[j] = '\0';
    printf("Sirul concatenat: %s\n", rezultat);
    return 0;
}
```

Output

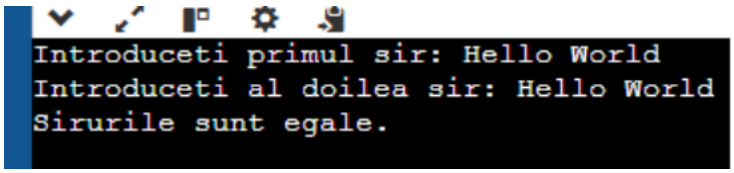
```
Introduceti primul sir: Hello world!
Introduceti al doilea sir:  Welcome!
Sirul concatenat: Hello world! Welcome!
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir1[100], sir2[100];

    printf("Introduceti primul sir: ");
    fgets(sir1, sizeof(sir1), stdin);

    printf("Introduceti al doilea sir: ");
    fgets(sir2, sizeof(sir2), stdin);
        int i = 0;
    while(sir1[i] != '\0' && sir1[i] != '\n') {
        i++;
    }
    sir1[i] = '\0';

    i = 0;
    while(sir2[i] != '\0' && sir2[i] != '\n') {
        i++;
    }
    sir2[i] = '\0';

    // Comparare
    i = 0;
    int egale = 1;
    while(sir1[i] != '\0' && sir2[i] != '\0') {
        if(sir1[i] != sir2[i]) {
            egale = 0;
            break;
        }
        i++;
    }
    if(egale && sir1[i] == sir2[i]) {
        printf("Sirurile sunt egale.\n");
    } else {
        printf("Sirurile NU sunt egale.\n");
    }

    return 0;
}
```

Output

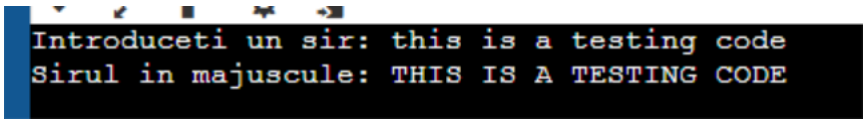


Input

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        int i = 0;
    while(sir[i] != '\0' && sir[i] != '\n') {
        i++;
    }
    sir[i] = '\0';
        i = 0;
    while(sir[i] != '\0') {
        if(sir[i] >= 'a' && sir[i] <= 'z') {
            sir[i] = sir[i] - 32;
        }
        i++;
    }

    printf("Sirul in majuscule: %s\n", sir);

    return 0;
}
```

Output



Input

```c
#include <stdio.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        int i = 0;
    while(sir[i] != '\0' && sir[i] != '\n') {
        i++;
    }
    sir[i] = '\0';
        i = 0;
    while(sir[i] != '\0') {
        if(sir[i] >= 'A' && sir[i] <= 'Z') {
            sir[i] = sir[i] + 32;
        }
        i++;
    }

    printf("Sirul in minuscule: %s\n", sir);

    return 0;
}
```

Output

```
Introduceti un sir: THIS IS A TESTING CODE
Sirul in minuscule: this is a testing code
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int i = 0;
    while(sir[i] != '\0' && sir[i] != '\n') {
        i++;
    }
    sir[i] = '\0';

    i = 0;
    while(sir[i] != '\0') {
        if(sir[i] >= 'a' && sir[i] <= 'z') {
            sir[i] = sir[i] - 32;
        } else if(sir[i] >= 'A' && sir[i] <= 'Z') {
            sir[i] = sir[i] + 32;
        }
        i++;
    }

    printf("Sirul cu litere comutate: %s\n", sir);

    return 0;
}
```

Output

```
Introduceti un sir: Welcome
Sirul cu litere comutate: wELCOME
```

Input

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int litere = 0, cifre = 0, speciale = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if((sir[i] >= 'a' && sir[i] <= 'z') ||
           (sir[i] >= 'A' && sir[i] <= 'Z')) {
            litere++;
        } else if(sir[i] >= '0' && sir[i] <= '9') {
            cifre++;
        } else if(sir[i] != ' ') {
            speciale++;
        }
        i++;
    }

    printf("Litere: %d\n", litere);
    printf("Cifre: %d\n", cifre);
    printf("Caractere speciale: %d\n", speciale);

    return 0;
}
```

Output

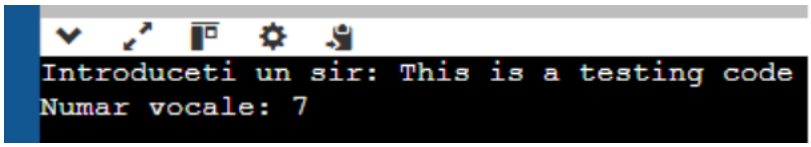
```
Introduceti un sir: Hello World 2!
Litere: 10
Cifre: 1
Caractere speciale: 1
```

Input

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int vocale = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        char c = tolower(sir[i]);
        if(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
            vocale++;
        }
        i++;
    }

    printf("Numar vocale: %d\n", vocale);

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int consoane = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        char c = tolower(sir[i]);
        if(c >= 'a' && c <= 'z') {
            if(c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u') {
                consoane++;
            }
        }
        i++;
    }

    printf("Numar consoane: %d\n", consoane);

    return 0;
}
```

Output

```
Introduceti un sir: This is a testing code
Numar consoane: 11
```

Input

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int cuvinte = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if(sir[i] != ' ' && (sir[i+1] == ' ' || sir[i+1] == '\0' || sir[i+1] == '\n')) {
            cuvinte++;
        }
        i++;
    }

    printf("Numar cuvinte: %d\n", cuvinte);

    return 0;
}
```

Output

```
Introduceti un sir: this is a testing program
Numar cuvinte: 5
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100], invers[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = 0;
    while(sir[lungime] != '\0' && sir[lungime] != '\n') {
        lungime++;
    }
    sir[lungime] = '\0';

    for(int i = 0; i < lungime; i++) {
        invers[i] = sir[lungime - 1 - i];
    }
    invers[lungime] = '\0';

    printf("Sirul original: %s\n", sir);
    printf("Sirul inversat: %s\n", invers);

    return 0;
}
```

Output

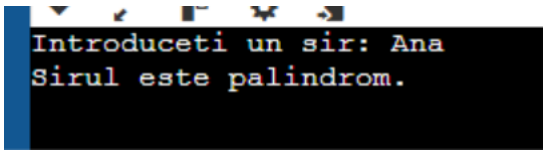
```
Introduceti un sir: Welcome
Sirul original: Welcome
Sirul inversat: emocleW
```

Input

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        int lungime = 0;
    while(sir[lungime] != '\0' && sir[lungime] != '\n') {
        lungime++;
    }
    sir[lungime] = '\0';
        int estePalindrom = 1;
    for(int i = 0; i < lungime/2; i++) {
        if(tolower(sir[i]) != tolower(sir[lungime - 1 - i])) {
            estePalindrom = 0;
            break;
        }
    }

    if(estePalindrom) {
        printf("Sirul este palindrom.\n");
    } else {
        printf("Sirul NU este palindrom.\n");
    }

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    char cuvinte[50][50];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        sir[strcspn(sir, "\n")] = '\0';
        int numarCuvinte = 0;
    int indexCuvant = 0;

    for(int i = 0; sir[i] != '\0'; i++) {
        if(sir[i] == ' ') {
            cuvinte[numarCuvinte][indexCuvant] = '\0';
            numarCuvinte++;
            indexCuvant = 0;
        } else {
            cuvinte[numarCuvinte][indexCuvant] = sir[i];
            indexCuvant++;
        }
    }
    cuvinte[numarCuvinte][indexCuvant] = '\0';
    numarCuvinte++;
        printf("Cuvintele in ordine inversa: ");
    for(int i = numarCuvinte - 1; i >= 0; i--) {
        printf("%s ", cuvinte[i]);
    }
    printf("\n");

    return 0;
}
```

Output

```
Introduceti un sir: Welcome world
Cuvintele in ordine inversa: world Welcome
```

Input

main.c

```c
#include <stdio.h
int main() {
    char sir[100], caracter;

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    printf("Introduceti caracterul de cautat: ");
    scanf("%c", &caracter);

    int pozitie = -1;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if(sir[i] == caracter) {
            pozitie = i;
            break;
        }
        i++;
    }

    if(pozitie != -1) {
        printf("Prima aparitie a caracterului '%c' este la pozitia: %d\n", caracter, pozitie);
    } else {
        printf("Caracterul '%c' nu a fost gasit in sir.\n", caracter);
    }

    return 0;
}
```

Output

```
Introduceti un sir: First Word
Introduceti caracterul de cautat: Word
Prima aparitie a caracterului 'W' este la pozitia: 6
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100], caracter;

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    printf("Introduceti caracterul de cautat: ");
    scanf("%c", &caracter);

    int pozitie = -1;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if(sir[i] == caracter) {
            pozitie = i;
        }
        i++;
    }

    if(pozitie != -1) {
        printf("Ultima aparitie a caracterului '%c' este la pozitia: %d\n", caracter, pozitie);
    } else {
        printf("Caracterul '%c' nu a fost gasit in sir.\n", caracter);
    }

    return 0;
}
```

Output

```
Introduceti un sir: Ultima litera este e
Introduceti caracterul de cautat: e
Ultima aparitie a caracterului 'e' este la pozitia: 19
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100], caracter;

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    printf("Introduceti caracterul de cautat: ");
    scanf("%c", &caracter);

    printf("Caracterul '%c' apare la pozitiile: ", caracter);
    int gasit = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if(sir[i] == caracter) {
            printf("%d ", i);
            gasit = 1;
        }
        i++;
    }

    if(!gasit) {
        printf("nicio pozitie");
    }
    printf("\n");

    return 0;
}
```

Output

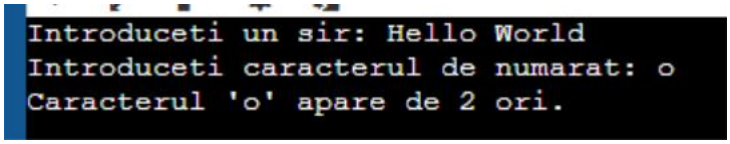
```
Introduceti un sir: Welcome World!123
Introduceti caracterul de cautat: W
Caracterul 'W' apare la pozitiile: 0 8
```

Input

main.c

```c
#include <stdio.h>

int main() {
    char sir[100], caracter;

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    printf("Introduceti caracterul de numarat: ");
    scanf("%c", &caracter);

    int count = 0;
    int i = 0;

    while(sir[i] != '\0' && sir[i] != '\n') {
        if(sir[i] == caracter) {
            count++;
        }
        i++;
    }

    printf("Caracterul '%c' apare de %d ori.\n", caracter, count);

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];

    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }
    int frecventa[256] = {0};
    int maxFrecventa = 0;
    char caracterMax;
        for(int i = 0; i < lungime; i++) {
        if(sir[i] != ' ') {
            frecventa[(int)sir[i]]++;
            if(frecventa[(int)sir[i]] > maxFrecventa) {
                maxFrecventa = frecventa[(int)sir[i]];
                caracterMax = sir[i];
            }
        }
    }
    printf("Caracterul cu cea mai mare frecventa este '%c' (%d aparitii).\n",
           caracterMax, maxFrecventa);

    return 0;
}
```

Output

```
Introduceti un sir: Welcome program this is a testing program
Caracterul cu cea mai mare frecventa este 'r' (4 aparitii).
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>
#include <limits.h>
int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
        int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }
    int frecventa[256] = {0};
    for(int i = 0; i < lungime; i++) {
        if(sir[i] != ' ') {
            frecventa[(int)sir[i]]++;
        }
    }
        int minFrecventa = INT_MAX;
    char caracterMin;

    for(int i = 0; i < 256; i++) {
        if(frecventa[i] > 0 && frecventa[i] < minFrecventa) {
            minFrecventa = frecventa[i];
            caracterMin = (char)i;
        }
    }
    if(minFrecventa != INT_MAX) {
        printf("Caracterul cu cea mai mica frecventa este '%c' (%d aparitii).\n",
                caracterMin, minFrecventa);
    } else {
        printf("Sirul este gol sau contine doar spatii.\n");
    }

    return 0;
```

Output

```
Introduceti un sir: Welcome this is a testing program
Caracterul cu cea mai mica frecventa este 'W' (1 aparitii).
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int frecventa[256] = {0};

    for(int i = 0; i < lungime; i++) {
        frecventa[(unsigned char)sir[i]]++;
    }

    printf("Frecventa caracterelor:\n");
    for(int i = 0; i < 256; i++) {
        if(frecventa[i] > 0) {
            if(i == ' ') {
                printf("Spatiu: %d\n", frecventa[i]);
            } else if(i == '\t') {
                printf("TAB: %d\n", frecventa[i]);
            } else {
                printf("'%c': %d\n", i, frecventa[i]);
            }
        }
    }

    return 0;
}
```

Output

```
Introduceti un sir: este un sir de test
Frecventa caracterelor:
Spatiu: 4
'd': 1
'e': 4
'i': 1
'n': 1
'r': 1
's': 3
't': 3
'u': 1
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char caracter;
    printf("Introduceti caracterul de eliminat: ");
    scanf("%c", &caracter);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int pozitie = -1;
    for(int i = 0; i < lungime; i++) {
        if(sir[i] == caracter) {
            pozitie = i;
            break;
        }
    }

    if(pozitie != -1) {
        for(int i = pozitie; i < lungime; i++) {
            sir[i] = sir[i+1];
        }
        printf("Sirul dupa eliminare: %s\n", sir);
    } else {
        printf("Caracterul nu a fost gasit in sir.\n");
    }

    return 0;
}
```

Output

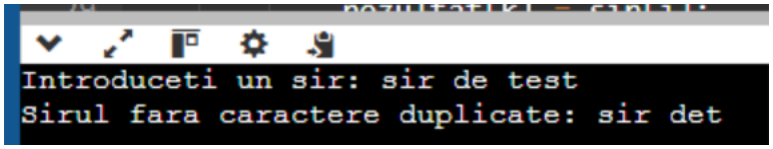
```
Introduceti un sir: Sir de test
Introduceti caracterul de eliminat: de
Sirul dupa eliminare: Sir e test
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    char rezultat[100];
    int k = 0;

    for(int i = 0; i < lungime; i++) {
        int gasit = 0;

        for(int j = 0; j < i; j++) {
            if(sir[i] == sir[j]) {
                gasit = 1;
                break;
            }
        }

        if(!gasit) {
            rezultat[k] = sir[i];
            k++;
        }
    }
    rezultat[k] = '\0';

    printf("Sirul fara caractere duplicate: %s\n", rezultat);

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char vechi, nou;
    printf("Introduceti caracterul de inlocuit: ");
    scanf(" %c", &vechi);
    printf("Introduceti caracterul nou: ");
    scanf(" %c", &nou);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int schimbat = 0;
    for(int i = 0; i < lungime; i++) {
        if(sir[i] == vechi && !schimbat) {
            sir[i] = nou;
            schimbat = 1;
        }
    }

    if(schimbat) {
        printf("Sirul dupa inlocuire: %s\n", sir);
    } else {
        printf("Caracterul nu a fost gasit in sir.\n");
    }

    return 0;
}
```

Output

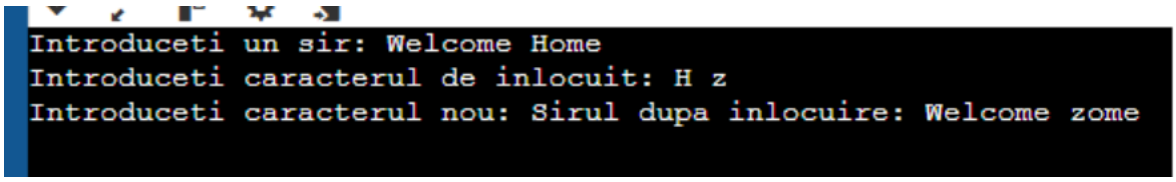
```
Introduceti un sir: Welcome
Introduceti caracterul de inlocuit: W e
Introduceti caracterul nou: Sirul dupa inlocuire: eelcome
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char vechi, nou;
    printf("Introduceti caracterul de inlocuit: ");
    scanf(" %c", &vechi);
    printf("Introduceti caracterul nou: ");
    scanf(" %c", &nou);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int pozitie = -1;
    for(int i = 0; i < lungime; i++) {
        if(sir[i] == vechi) {
            pozitie = i;
        }
    }

    if(pozitie != -1) {
        sir[pozitie] = nou;
        printf("Sirul dupa inlocuire: %s\n", sir);
    } else {
        printf("Caracterul nu a fost gasit in sir.\n");
    }

    return 0;
}
```

Output



Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de cautat: ");
    fgets(cuvant, sizeof(cuvant), stdin);

    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
    }

    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }

    char *pozitie = strstr(sir, cuvant);

    if(pozitie != NULL) {
        int index = pozitie - sir;
        printf("Prima aparitie a cuvantului '%s' este la pozitia: %d\n", cuvant, index);
    } else {
        printf("Cuvantul '%s' nu a fost gasit in sir.\n", cuvant);
    }

    return 0;
}
```

Output

```
Introduceti un sir: este un sir de test
Introduceti cuvantul de cautat: un
Prima aparitie a cuvantului 'un' este la pozitia: 5
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de cautat: ");
    fgets(cuvant, sizeof(cuvant), stdin);

    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
    }
    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }
    char *ultima_pozitie = NULL;
    char *curent = sir;
    while((curent = strstr(curent, cuvant)) != NULL) {
        ultima_pozitie = curent;
        curent++;
    }

    if(ultima_pozitie != NULL) {
        int index = ultima_pozitie - sir;
        printf("Ultima aparitie a cuvantului '%s' este la pozitia: %d\n", cuvant, index);
    } else {
        printf("Cuvantul '%s' nu a fost gasit in sir.\n", cuvant);
    }

    return 0;
}
```

Output

```
Introduceti un sir: este un sir de test
Introduceti cuvantul de cautat: de
Ultima aparitie a cuvantului 'de' este la pozitia: 12
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de cautat: ");
    fgets(cuvant, sizeof(cuvant), stdin);
    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
    }
    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }
    printf("Cuvantul '%s' apare la pozitiile: ", cuvant);
    char *curent = sir;
    int gasit = 0;
    while((curent = strstr(curent, cuvant)) != NULL) {
        int index = curent - sir;
        printf("%d ", index);
        gasit = 1;
        curent++;
    }
    if(!gasit) {
        printf("nicio pozitie");
    }
    printf("\n");
    return 0;
}
```

Output

```
Introduceti un sir: test acesta este un sir de test
Introduceti cuvantul de cautat: test
Cuvantul 'test' apare la pozitiile: 0 27
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de numarat: ");
    fgets(cuvant, sizeof(cuvant), stdin);
    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
    }

    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }

    int count = 0;
    char *curent = sir;

    while((curent = strstr(curent, cuvant)) != NULL) {
        count++;
        curent += lungime_cuvant;
    }

    printf("Cuvantul '%s' apare de %d ori.\n", cuvant, count);

    return 0;
}
```

Output

```
Introduceti un sir: hello world!
Introduceti cuvantul de numarat: l
Cuvantul 'l' apare de 3 ori.
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de eliminat: ");
    fgets(cuvant, sizeof(cuvant), stdin);

    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
        lungime_sir--;
    }

    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }

    char *pozitie = strstr(sir, cuvant);

    if(pozitie != NULL) {
        int index = pozitie - sir;

        for(int i = index; i <= lungime_sir - lungime_cuvant; i++) {
            sir[i] = sir[i + lungime_cuvant];
        }

        printf("Sirul dupa eliminare: %s\n", sir);
    } else {
        printf("Cuvantul nu a fost gasit in sir.\n");
    }
    return 0;
}
```

Output

```
Introduceti un sir: hello world
Introduceti cuvantul de eliminat: world
Sirul dupa eliminare: hello
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de eliminat: ");
    fgets(cuvant, sizeof(cuvant), stdin);

    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
        lungime_sir--;
    }
    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }
    char *ultima_pozitie = NULL;
    char *curent = sir;

    while((curent = strstr(curent, cuvant)) != NULL) {
        ultima_pozitie = curent;
        curent++;
    }
    if(ultima_pozitie != NULL) {
        int index = ultima_pozitie - sir;

        for(int i = index; i <= lungime_sir - lungime_cuvant; i++) {
            sir[i] = sir[i + lungime_cuvant];
        }

        printf("Sirul dupa eliminare: %s\n", sir);
    } else {
        printf("Cuvantul nu a fost gasit in sir.\n");
    }

    return 0;
}
```

Output

```
Introduceti un sir: este un sir de de test
Introduceti cuvantul de eliminat: de
Sirul dupa eliminare: este un sir de  test
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    char cuvant[50];
    printf("Introduceti cuvantul de eliminat: ");
    fgets(cuvant, sizeof(cuvant), stdin);

    int lungime_sir = strlen(sir);
    if(sir[lungime_sir-1] == '\n') {
        sir[lungime_sir-1] = '\0';
        lungime_sir--;
    }

    int lungime_cuvant = strlen(cuvant);
    if(cuvant[lungime_cuvant-1] == '\n') {
        cuvant[lungime_cuvant-1] = '\0';
        lungime_cuvant--;
    }

    char rezultat[200] = "";
    char *cuvant_curent = strtok(sir, " ");
    int prima = 1;

    while(cuvant_curent != NULL) {
        if(strcmp(cuvant_curent, cuvant) != 0) {
            if(!prima) {
                strcat(rezultat, " ");
            }
            strcat(rezultat, cuvant_curent);
            prima = 0;
        }
        cuvant_curent = strtok(NULL, " ");
    }

    printf("Sirul dupa eliminare: %s\n", rezultat);

    return 0;
}
```

Output

```
Introduceti un sir: acesta este un sir de test, sir rar
Introduceti cuvantul de eliminat: sir
Sirul dupa eliminare: acesta este un de test, rar
```

Input

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int start = 0;
    while(isspace(sir[start])) {
        start++;
    }
        for(int i = 0; i < lungime - start; i++) {
        sir[i] = sir[i + start];
    }
    sir[lungime - start] = '\0';

    printf("Sirul fara spatii la inceput: '%s'\n", sir);

    return 0;
}
```

Output

```
Introduceti un sir:                    este un sir de test
Sirul fara spatii la inceput: 'este un sir de test'
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }

    int end = lungime - 1;
    while(end >= 0 && isspace(sir[end])) {
        end--;
    }

    sir[end + 1] = '\0';

    printf("Sirul fara spatii la sfarsit: '%s'\n", sir);

    return 0;
}
```

Output

```
Introduceti un sir: este un sir de test
Sirul fara spatii la sfarsit: 'este un sir de test'
```

Input

main.c

```c
#include <string.h>
#include <ctype.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }
        int start = 0;
    while(isspace(sir[start])) {
        start++;
    }
        int end = lungime - 1;
    while(end >= start && isspace(sir[end])) {
        end--;
    }
        char rezultat[100];
    int j = 0;
    for(int i = start; i <= end; i++) {
        rezultat[j] = sir[i];
        j++;
    }
    rezultat[j] = '\0';

    printf("Sirul dupa trimitere: '%s'\n", rezultat);

    return 0;
}
```

Output

```
Introduceti un sir:    este un sir de test
Sirul dupa trimitere: 'este un sir de test'
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);
    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }
        char litere[100];
    int k = 0;
    for(int i = 0; i < lungime; i++) {
        if(isalpha(sir[i])) {
            litere[k] = sir[i];
            k++;
        }
    }
    litere[k] = '\0';
        for(int i = 0; i < k-1; i++) {
        for(int j = 0; j < k-i-1; j++) {
            if(tolower(litere[j]) > tolower(litere[j+1])) {
                char temp = litere[j];
                litere[j] = litere[j+1];
                litere[j+1] = temp;
            }
        }
    }
        k = 0;
    for(int i = 0; i < lungime; i++) {
        if(isalpha(sir[i])) {
            sir[i] = litere[k];
            k++;
        }
    }
    printf("Sirul cu litere sortate: %s\n", sir);

    return 0;
}
```

Output

```
Introduceti un sir: Sir de test
Sirul cu litere sortate: dee ir Sstt
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char sir[100];
    printf("Introduceti un sir: ");
    fgets(sir, sizeof(sir), stdin);

    int lungime = strlen(sir);
    if(sir[lungime-1] == '\n') {
        sir[lungime-1] = '\0';
        lungime--;
    }
        char cuvinte[50][50];
    int numar_cuvinte = 0;
    char *cuvant = strtok(sir, " ");

    while(cuvant != NULL && numar_cuvinte < 50) {
        strcpy(cuvinte[numar_cuvinte], cuvant);
        numar_cuvinte++;
        cuvant = strtok(NULL, " ");
    }
        for(int i = 0; i < numar_cuvinte-1; i++) {
        for(int j = 0; j < numar_cuvinte-i-1; j++) {
            if(strcasecmp(cuvinte[j], cuvinte[j+1]) > 0) {
                char temp[50];
                strcpy(temp, cuvinte[j]);
                strcpy(cuvinte[j], cuvinte[j+1]);
                strcpy(cuvinte[j+1], temp);
            }
        }
    }
    printf("Cuvintele sortate: ");
    for(int i = 0; i < numar_cuvinte; i++) {
        printf("%s ", cuvinte[i]);
    }
    printf("\n");

    return 0;
}
```

Output

```
Introduceti un sir: sir de test
Cuvintele sortate: de sir test
```

Input

Output

```
Introduceti o expresie matematica: (2*5)+7-2
Expresie matematica corecta.
```

Input

main.c

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main() {
    char expresie[100];
    printf("Introduceti o expresie matematica: ");
    fgets(expresie, sizeof(expresie), stdin);
    int lungime = strlen(expresie);
    if(expresie[lungime-1] == '\n') {
        expresie[lungime-1] = '\0';
        lungime--;}
    int paranteze = 0;
    int corecta = 1;
    for(int i = 0; i < lungime; i++) {
        char c = expresie[i];
        if(!isdigit(c) &&
           c != '+' && c != '-' && c != '*' && c != '/' &&
           c != '(' && c != ')' && c != ' ') {
            corecta = 0;
            break;}}
    if(corecta)  {       for(int i = 0; i < lungime; i++) {
            if(expresie[i] == '(') {
                paranteze++;
            } else if(expresie[i] == ')') {
                paranteze--;
                if(paranteze < 0) {
                    corecta = 0;
                    break;}}}
                for(int i = 0; i < lungime-1; i++) {
            char c1 = expresie[i];
            char c2 = expresie[i+1];

            if((c1 == '+' || c1 == '-' || c1 == '*' || c1 == '/') &&
               (c2 == '+' || c2 == '-' || c2 == '*' || c2 == '/')) {
                corecta = 0;
                break;
            }
                        if(i == 0 && (c1 == '*' || c1 == '/' || c1 == ')')) {
                corecta = 0;
                break;}}
                char ultim = expresie[lungime-1];
        if(ultim == '+' || ultim == '-' || ultim == '*' || ultim == '/' || ultim == '(') {
            corecta = 0;}}
        if(paranteze != 0) {
        corecta = 0;}
    if(corecta) {
        printf("Expresie matematica corecta.\n");} else {
        printf("Expresie matematica gresita.\n");}
    return 0;}
```

Output

```
Introduceti o expresie matematica: (2*5)+7-2
Expresie matematica corecta.
```