**Реализация алгоритма манипулирования данными**

#include <stdio.h>

#include <stdlib.h>

int Char\_to\_Int(char ch){

int result=0;

if (ch == ' ') ch = getchar();

while (ch != ' ' && ch != ',' && ch != '.'){

if (ch >= '0' && ch <= '9' )

result = result\*10 + (ch - '0');

if (ch == '-') printf("%c", ch);

ch = getchar();

}

return result;

}

int main(int argc, const char \* argv[]){

srand(time(NULL));

int N = 4 + rand() % 5;

int M = 5 + rand() % 5;

int N1, N2, M1, M2;

int i, j;

int i1, j1;

int mul, x;

char ch;

int \*\*arr1;

int \*\*arr\_min;

int \*\*arr2;

int row;

int min = 100;

int i\_min = 0, j\_min = 0;

arr1 = (int\*\*) malloc(N\*sizeof(int\*));

for (i = 0; i < N; i++)

arr1[i] = (int\*) malloc(M \* sizeof(int));

if (arr1 == NULL){

printf ("Memory allocation error.");

return 0;

}

printf("Random array:\n"); // Задаем произвольный массив

for (i = 0; i < N; i++){

for (j = 0; j < M; j++){

arr1[i][j] = 10 + rand() % 90;

printf("%d ", arr1[i][j]);

}

printf("\n");

}

// меняем местами первую и последнюю строки, развернув их

printf("\nReplacement of the 1st row to the last one.\n");

for(j = 0; j < M; j++){

row = arr1[0][j];

arr1[0][j] = arr1[N-1][M-j-1];

arr1[N-1][M-1-j] = row;

}

for (i = 0; i < N; i++){

for (j = 0; j < M; j++)

printf("%d ", arr1[i][j]);

printf("\n");

}

// Определяем наименьший элемент матрицы

for (i = 0; i < N; i ++){

for (j = 0; j < M; j++){

if (arr1[i][j] < min){

min = arr1[i][j];

i\_min = i;

j\_min = j;

}

}

}

arr\_min = (int\*\*) malloc((N-1)\*sizeof(int\*));

for (i = 0; i < N; i++)

arr\_min[i] = (int\*) malloc((M-1) \* sizeof(int));

if (arr\_min == NULL){

printf ("\nMemory allocation error.");

return 0;

}

//Удаляем строку и столбец, на пересечении которых наименьший элемент

for (i = 0, i1=0; i < N; i ++){

if (i != i\_min){

for (j = 0, j1=0; j < M; j++){

if (j != j\_min) {

arr\_min[i1][j1] = arr1[i][j];

j1++;

}

}

i1++;

}

}

printf("\nWithout row and column with min element:\n");

for (i1 = 0; i1 < N-1; i1++){

for (j1 = 0; j1 < M-1; j1++)

printf("%d ", arr\_min[i1][j1]);

printf("\n");

}

N1 = N-1;

M1 = M-1;

//Задаем размер второй матрицы

printf("\nEnter the size of 2nd matrix: ");

scanf("%d %d", &N2, &M2);

// Проверяем можно ли перемножать матрицы

if (M1 != N2){

printf("\nMatrices can't be multiplied\n");

return 0;

}

arr2 = (int\*\*) malloc(N2\*sizeof(int\*));

for (i = 0; i < N2; i++)

arr2[i] = (int\*) malloc(M2 \* sizeof(int));

if (arr2 == NULL){

printf ("Memory allocation error.");

return 0;

}

// Ввод элементов второй матрицы

printf("\nEnter 2nd matrix's elements\n");

for (i = 0; i< N2; i++){

for(j = 0; j < M2; j++){

ch = getchar();

x = Char\_to\_Int(ch);

if(x != 0) arr2[i][j] = x;

}

}

printf("\n2nd matrix:\n");

for (i = 0; i < N2; i++){

for (j = 0; j < M2; j++)

printf("%d ", arr2[i][j]);

printf("\n");

}

// Перемножаем полученные матрицы

int C[N1][M2];

for(i1 = 0; i1 < N1; i1++)

for(j = 0; j < M2; j++){

C[i1][j] = 0;

for(j1 = 0; j1 < M1; j1++)

C[i1][j] += arr\_min[i1][j1] \* arr2[j1][j];

}

printf("\nResult of multiplying:\n");

for (i1 = 0; i1 < N1; i1++){

for (j=0; j < M2; j++)

printf("%d ", C[i1][j]);

printf("\n");

}

return 0;

}