Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

**№4**

з дисципліни:

«Інформатика1. Основи програмування та алгоритми»

|  |  |
| --- | --- |
|  | Виконала: Петруненко Олександр  Група: РЕ–12  Викладач: доцент Катін Ю.П. |

Київ – 2021

Мета роботи: навчитись працювати з двовимірними масивами, адресами та вказівниками.

Код:

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <locale.h>**

**#include <conio.h>**

**#include <malloc.h>**

**int i,j,temp,N=3,M=4,choise, v ;**

**int maximal=0,sort,sum=0,vvod,end;**

**int Maximum(int \*\*A)**

**{**

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

**{**

**for(j=0;j<N;j++)**

**{**

**if (A[i][j] > maximal)**

**{**

**maximal = A[i][j];**

**}**

**}**

**}**

**printf("\nМаксимальное значение матрици А: %d\n",maximal);**

**}**

**int Transponate (int \*\*B,int \*\*T)**

**{**

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

**{**

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

**{**

**T[j][i] = B[i][j];**

**}**

**}**

**printf("\n Транспонированая матрица B : \n");**

**for (i=0;i<M;i++)**

**{**

**for (j=0;j<N;j++)**

**{**

**printf("%3d ", T[i][j]);**

**}**

**printf("\n");**

**}**

**}**

**int Multiply(int \*\*A,int \*\*B,int \*\*Dob)**

**{**

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

**{**

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

**{**

**Dob[i][j]=0;**

**for(int p=0;p<N;p++)**

**{**

**Dob[i][j] += A[i][p] \* B[p][j];**

**}**

**}**

**}**

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

**{**

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

**{**

**printf("%3d ", Dob[i][j]);**

**}**

**printf("\n");**

**}**

**}**

**int Sort(int \*\*A,int sort)**

**{**

**for(int k=1;k<N;k++)**

**{**

**for(j=0;j<N-k;j++)**

**{**

**if(A[sort][j]>A[sort][j+1])**

**{**

**temp = A[sort][j];**

**A[sort][j] = A[sort][j+1];**

**A[sort][j+1] = temp;**

**}**

**}**

**}**

**for(j=0;j<N;j++)**

**{**

**printf("%3d ",A[sort][j]);**

**}**

**}**

**int Summ(int \*\*A,int \*\*B,int sum)**

**{**

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

**{**

**sum=0;**

**for(j=0;j<N;j++)**

**{**

**sum += A[i][j];**

**}**

**printf("\n%d Сумма элементов строк матрицы А =%d",i,sum);**

**}**

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

**{**

**sum=0;**

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

**{**

**sum += B[i][j];**

**}**

**printf("\n%d Сумма элементов столбцов матрицы В =%d",j,sum);**

**}**

**}**

**int main()**

**{ setlocale(LC\_ALL, "Rus");**

**for(int q=0;q<5;q++)**

**{**

**printf("Доброго времени суток))))\n");**

**printf("Выберите создание матрицы:\n1. Автоматическое\n2. С клавиатуры\n");**

**scanf("%d",&vvod);**

**if (vvod==2)**

**{**

**printf("\nВведите размер квадратной матрицы: ");**

**scanf("%d",&N);**

**printf("\nВведите размер прямоугольной матрицы: ");**

**scanf("%d",&M);**

**}**

**int \*\*A=NULL;**

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

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

**{**

**A[i]=(int\*) malloc(N \* sizeof(int));**

**}**

**int \*\*B=NULL;**

**B=(int\*\*) malloc(M \* sizeof(int));**

**int \*\*Dob=NULL;**

**Dob=(int\*\*) malloc(M \* sizeof(int));**

**int \*\*T=NULL;**

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

**for(int i=0;i<M;i++)**

**{**

**B[i]=(int\*) malloc(N \* sizeof(int));**

**Dob[i]=(int\*) malloc(N \* sizeof(int));**

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

**}**

**if(vvod==2)**

**{**

**printf("Введите элементы квадратной матрицы:\n");**

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

**{**

**for (j=0;j<N;j++)**

**{**

**printf("A[%d][%d]=",i,j);**

**scanf("%d",&A[i][j]);**

**}**

**}**

**printf("Введите элементы прямоугольной матрицы:\n");**

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

**{**

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

**{**

**printf("B[%d][%d]=",i,j);**

**scanf("%d",&B[i][j]);**

**}**

**}**

**}**

**else**

**{**

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

**{**

**for (j=0;j<N;j++)**

**{**

**A[i][j]= rand() %10;**

**}**

**}**

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

**{**

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

**{**

**B[i][j]= rand() %15;**

**}**

**}**

**}**

**printf("\nИсходная матрица A и B:\n");**

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

**{**

**for (j=0;j<N;j++)**

**{**

**printf("%3d ", A[i][j]);**

**}**

**printf("\n");**

**}**

**printf("\n");**

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

**{**

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

**{**

**printf("%3d ", B[i][j]);**

**}**

**printf("\n");**

**}**

**do**

**{**

**printf("\nМеню управления::\n1. Мах А \n2. Транспонировать B\n3. Умножить AxB\n4. Сортировать строку A\n5. Показать сумму в строках и столбцах\n");**

**scanf("%d",&choise);**

**}while(choise != 1 && choise != 2 && choise != 3 && choise != 4 && choise != 5);**

**switch (choise)**

**{**

**case 1:**

**Maximum(A);**

**break;**

**case 2:**

**Transponate(B,T);**

**break;**

**case 3:**

**printf("AxB:\n");**

**Multiply(A,B,Dob);**

**break;**

**case 4:**

**printf("\nВведите номер строки для сортировки: ");**

**scanf("%d",&sort);**

**Sort(A,sort);**

**break;**

**case 5:**

**Summ(A,B,sum);**

**break;**

**}**

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

**{**

**free(A[i]);**

**}**

**free(A);**

**for(int i=0;i<M;i++)**

**{**

**free(B[i]);**

**free(Dob[i]);**

**free(T[i]);**

**}**

**free(B);**

**free(Dob);**

**free(T);**

**printf("\n\n");**

**}**

**return 0;**

**}**

**Висновок:**

**Матриці А та B (A квадратна, B прямокутна)**

**Ми написали програми, що виконують певні дії с масивами – множення, транспонування, сортування, знаходження найбільшого числа у певній строчці.**

**Для динамічного масиву ми виділяємо пам'ять, яку маємо очистити її після виконання програми.**













