**Лабораторная работа 5  
Статические массивы**

**Цель работы**: изучить работу статических массивов в языке программирования С.

**Задание 1**

**Постановка задачи:**

Для некоторого числового вектора X, введённого с клавиатуры, вычислить значения вектора Y = X · X (yi = xi · xi — поэлементно).

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define size 10

int main(){

int x[size];

for (int i = 0; i < size; i++) {

scanf("%i", &x[i]);

x[i] = x[i] \* x[i];

}

for (int i = 0; i < size; i++)

printf("%i ", x[i]);

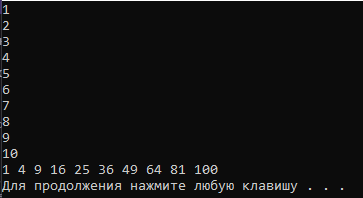
printf("\n");

system("pause");

return 0;

}

**Вывод программы:**



**Задание 2**

**Постановка задачи:**

Для некоторого числового массива X, введённого с клавиатуры, изменить порядок элементов на обратный ему без привлечения вспомогательного массива и со вспомогательным массивом

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define size 10

int x[size];

int rex[size];

void showx();

int main() {

for (int i = 0; i < size; i++) {

scanf("%d", &x[i]);

rex[size - 1 - i] = x[i];

}

showx();

for (int i = 0; i < size; i++)

x[i] = rex[i];

showx();

for (int i = 0; i < size / 2; i++) {

int temp = x[i];

x[i] = x[size - 1 - i];

x[size - 1 - i] = temp;

}

showx();

system("pause");

return 0;

}

void showx (){

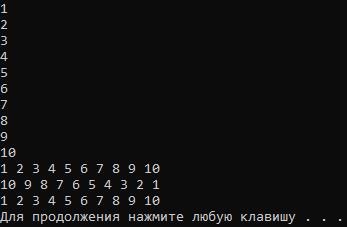
for (int i = 0; i < size; i++)

printf("%d ", x[i]);

printf("\n");

}

**Вывод программы:**



**Задание 3**

**Постановка задачи:**

Реализовать различные варианты алгоритма сортировки пузырьком, организовав проходы алгоритма с начала, и с конца массива, а также с двумя противоположными условиями сравнения. В качестве элементов сортировки использовать произвольные массивы чисел. Каждый найденный возможный вариант алгоритма должен приводить к некоторому осмысленному результату сортировки.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define size 10

int arr[size];

void show( int nmb );

void newArr();

int main() {

srand(time(NULL));

newArr();

show(1);

for (int i = 0; i < size; i++) {

for (int j = i + 1; j < size; j++) {

if (arr[j] < arr[i]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

show(0);

newArr();

show(2);

for (int i = 0; i < size; i++) {

for (int j = i + 1; j < size; j++) {

if (arr[i] < arr[j]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

show(0);

newArr();

show(3);

for (int i = size - 1; i > 0; i--) {

for (int j = i - 1; j >= 0; j--) {

if (arr[j] < arr[i]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

newArr();

show(4);

for (int i = size - 1; i > 0; i--) {

for (int j = i - 1; j >= 0; j--) {

if (arr[i] < arr[j]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

show(0);

system("pause");

return 0;

}

void show( int nmb ) {

if ( nmb != 0)

printf("%i)", nmb);

for (int i = 0; i < size; i++)

printf("%d ", arr[i]);

printf("\n\n");

}

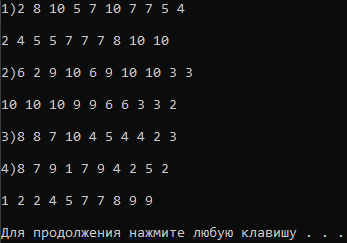
void newArr() {

for (int i = 0; i < size; i++)

arr[i] = rand() % 10 + 1;

}

**Вывод программы:**



**Задание 4**

**Постановка задачи:**

Реализовать самостоятельно алгоритм сортировки вставками и сравнить его реализацию с полученными реализациями «алгоритма пузырька» в задаче 5.3.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define size 10

int arr[size];

void show( int nmb );

void newArr();

int main() {

newArr();

show(1);

for (int i = 0; i < size; i++) { // Ascending

int index = i;

for (int j = i + 1; j < size; j++) {

if (arr[index] > arr[j])

index = j;

}

if (index != i) {

int temp = arr[index];

arr[index] = arr[i];

arr[i] = temp;

}

}

show(0);

newArr();

show(2);

for (int i = 0; i < size; i++) { // Descending

int index = i;

for (int j = i + 1; j < size; j++) {

if (arr[index] < arr[j])

index = j;

}

if (index != i) {

int temp = arr[index];

arr[index] = arr[i];

arr[i] = temp;

}

}

show(0);

system("pause");

return 0;

}

void show( int nmb ) {

if ( nmb != 0)

printf("%i)", nmb);

for (int i = 0; i < size; i++)

printf("%d ", arr[i]);

printf("\n\n");

}

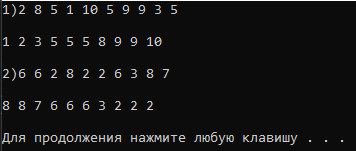
void newArr() {

for (int i = 0; i < size; i++)

arr[i] = rand() % 10 + 1;

}

**Вывод программы:**



**Задание 5**

**Постановка задачи:** Организовать ввод массива (матрицы) по столбцам

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define rows 2

#define columns 4

int arr[rows][columns];

int main() {

int\* fEl = &arr[0][0];

for (int column = 0; column < columns; column++) {

for (int row = 0; row < rows; row++) {

scanf("%d", fEl + column + row \* columns);

}

}

for (int row = 0; row < rows; row++) {

for (int column = 0; column < columns; column++) {

printf("%d ", arr[row][column]);

}

printf("\n");

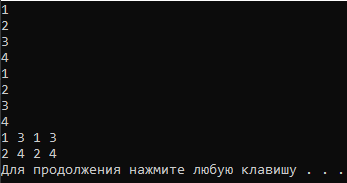
}

system("pause");

return 0;

}

**Вывод программы:**



**Задание 6**

**Постановка задачи:**

Найти значения матричного многочлена f(A), где 

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define m 2

#define n 2

int main() {

int k = 1;

int A[m][n] ;

printf("Matrix A:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

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

}

}

int AA[m][n];

printf("matrix AA:\n");

for(int i = 0; i < m; i++) {

for(int j = 0; j < n; j++) {

AA[i][j] = 0;

for(int r = 0; r < m; r++){

AA[i][j] += A[i][r] \* A[r][j];

}

printf("%i ", AA[i][j]);

}

printf("\n");

}

int E[m][n] = {{1,0}, {0,1}};

printf("matrix f(A):\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

A[i][j] = -2 \* AA[i][j] + 5 \* A[i][j] + 9 \* E[i][j];

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

}

printf("\n");

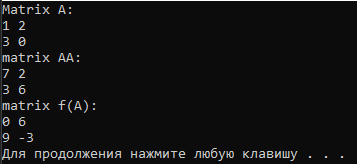
}

system(" pause");

return 0;

}

**Вывод программы:**



**Задание 7**

**Постановка задачи:**

Транспонировать матрицу 

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define m 2

#define n 3

int main() {

int A[m][n] = {{1,2,3}, {4,5,6}};

int AT[n][m];

printf("transp matrix A:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

AT[i][j] = A[j][i];

printf("%i ", AT[i][j]);

}

printf("\n");

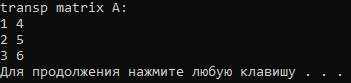
}

system(" pause");

return 0;

}

**Вывод программы:**



**Задание 8**

**Постановка задачи:**

Вычислить произведения A × AT и AT × A при заданной матрице A: 

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define m 2

#define n 4

int main() {

int A[m][n] = {{1, 2, 1, 3}, {4, -1, 5, -1}};

int AT[n][m];

printf("transp matrix A:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

AT[i][j] = A[j][i];

printf("%i ", AT[i][j]);

}

printf("\n");

}

int AAT[m][m];

printf("matrix AAT:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < m; j++) {

AAT[i][j] = 0;

for (int r = 0; r < n; r++) {

AAT[i][j] += A[i][r] \* AT[r][j];

}

printf("%i ", AAT[i][j]);

}

printf("\n");

}

int ATA[n][n];

printf("matrix ATA:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

ATA[i][j] = 0;

for (int r = 0; r < m; r++) {

ATA[i][j] += AT[i][r] \* A[r][j];

}

printf("%i ", ATA[i][j]);

}

printf("\n");

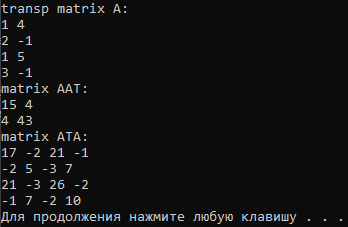
}

system(" pause");

return 0;

}

**Вывод программы:**



**Задание 9**

**Постановка задачи:**

Найти произведения матриц (AB) · C и A · (BC) 

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define n 2

int main() {

int A[n][n] = {{1, -1}, {-1, 1}};

int B[n][n] = {{2, 0}, {-3, 1}};

int C[n][n] = {{3, -1}, {2, 3}};

int AB[n][n];

printf("matrix AB:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

AB[i][j] = 0;

for (int r = 0; r < n; r++) {

AB[i][j] += A[i][r] \* B[r][j];

}

printf("%i ", AB[i][j]);

}

printf("\n");

}

int AB\_C[n][n];

printf("matrix (AB)C:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

AB\_C[i][j] = 0;

for (int r = 0; r < n; r++) {

AB\_C[i][j] += AB[i][r] \* C[r][j];

}

printf("%i ", AB\_C[i][j]);

}

printf("\n");

}

int BC[n][n];

printf("matrix BC:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

BC[i][j] = 0;

for (int r = 0; r < n; r++) {

BC[i][j] += B[i][r] \* C[r][j];

}

printf("%i ", BC[i][j]);

}

printf("\n");

}

int A\_BC[n][n];

printf("matrix A(BC):\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

A\_BC[i][j] = 0;

for (int r = 0; r < n; r++) {

A\_BC[i][j] += A[i][r] \* BC[r][j];

}

printf("%i ", A\_BC[i][j]);

}

printf("\n");

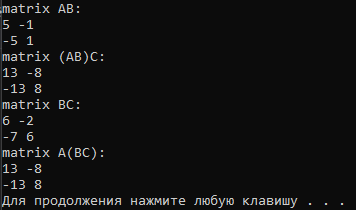
}

system(" pause");

return 0;

}

**Вывод программы:**



**Задание 10**

**Постановка задачи:**

Преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменён средним арифметическим элементов этой строки.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#define m 4

#define n 3

int main() {

int A[m][n] = {{4, 2, 3}, {5, 6, 7}, {2, 4, 6}, {2, 0, 7}};

printf("matrix A:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

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

printf("\n");

}

for (int i = 0; i < m; i++) {

for (int j = 1; j < n; j++) {

A[i][0] += A[i][j];

}

A[i][0] /= n ;

}

printf("matrix A after processing:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

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

printf("\n");

}

system(" pause");

return 0;

}

**Вывод программы:**

