Лабораторная работа №5

Массивы

**Задание 5.1**

**Постановка задачи:**

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

**Математическая модель:**

Представим вектор как одномерный массив чисел. Для вычисления значения Y нужно возвести в квадрат каждый элемент.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| a[5] | Вектор Х | int |
| i | Переменная, позволяющая работать с элементами массива | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

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

int a[5];

int i;

printf("Enter X:\n" );

for (i = 0; i<5; i++)

{

printf("a[%d] = ", i);

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

a[i]=a[i]\*a[i];

}

printf("\n");

printf("X^2:\n");

for (i = 0;i<5; i++)

{

printf("a[%d] = %d\n",i,a[i]);

}

return 0;

}#include <stdio.h>

#include <stdlib.h>

#include <math.h>

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

int a[5];

int i;

printf("Enter X:\n" );

for (i = 0; i<5; i++)

{

printf("a[%d] = ", i);

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

a[i]=a[i]\*a[i];

}

printf("\n");

printf("X^2:\n");

for (i = 0;i<5; i++)

{

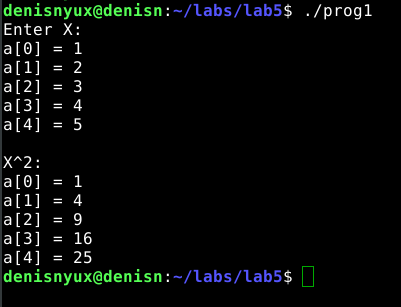
printf("a[%d] = %d\n",i,a[i]);

}

return 0;

}

**Результат выполненной работы:**



**Задание 5.2**

**Постановка задачи:**

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

**Математическая модель:**

Вводится массив, затем в цикле от i до i/2 каждый элемент i меняется на n-i-1, таким образом значения в массиве, находящиеся на одном и том же расстоянии от элемента массива n/2 меняются местами.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n | Количество элементов в массиве | int |
| a[n] | Массив | int |
| i | Переменная, позволяющая работать с элементами массива | int |
| t | Временная переменная для того чтобы менять значения элементов в цикле | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 6;

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

{

int i,t;

int a[n];

/\*Ввод массива\*/

printf("Enter X:\n" );

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

{

printf("a[%d] = ", i);

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

}

printf("\n");

/\*Каждый элемент i меняется на элемент n-i-1\*/

for (i=0; i<(n/2);i++)

{

t=a[n-i-1];

a[n-i-1]=a[i];

a[i]=t;

}

printf("\n");

/\*Вывод массива\*/

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

{

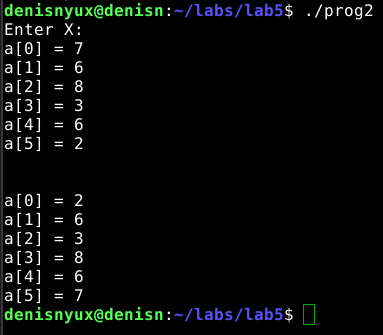
printf("a[%d] = %d\n",i,a[i]);

}

return 0;

}

**Результат выполненной работы:**



**Задание 5.3**

**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.

**Математическая модель:**

Алгоритм состоит из повторяющихся проходов по массиву. За каждый проход элементы сравниваются попарно и, если порядок в паре не соответствует условию, выполняется обмен элементов.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n | Количество элементов в массивах | int |
| a[n], b[n] | Массивы | int |
| i, j | Переменная, позволяющая работать с элементами массивов | int |
| t | Временная переменная для того чтобы менять значения элементов в цикле | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 8;

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

{

int i,j,t;

int a[n],b[n];

/\*Отсортируем массив А по возрастанию, проходя по массиву с начала,

а массив B отсортируем по убыванию, проходя по массиву с конца\*/

/\*Ввод массивов\*/

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

{

a[i]=rand()%20;

b[i]=rand()%20;

printf("a[%d] = %d\t b[%d]=%d\n", i, a[i],i,b[i]);

}

printf("\n");

/\*Сортировка массива А\*/

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

{

for (j=i+1;j<n;j++)

{

if (a[i]<=a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

/\*Сортировка массива B\*/

for (i=n-1; i>=0; i--)

{

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

{

if (b[i]>=b[j])

{

t=b[i];

b[i]=b[j];

b[j]=t;

}

}

}

/\*Вывод массива\*/

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

{

printf("a[%d] = %d\t b[%d]=%d\n", i, a[i],i,b[i]);

}

return 0;

}

**Результат выполненной работы:**

****

**Задание 5.4**

**Постановка задачи:**

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

**Математическая модель:**

Проходим по массиву слева направо и обрабатываем по очереди каждый элемент. Слева от очередного элемента наращиваем отсортированную часть массива. В отсортированной части массива ищется точка вставки для очередного элемента.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n | Количество элементов в массивах | int |
| a[n], b[n] | Массивы | int |
| i, j | Переменная, позволяющая работать с элементами массивов | int |
| t | Временная переменная для того чтобы менять значения элементов в цикле | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 10;

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

{

int i,j,t;

int a[n],b[n];

/\*Ввод массивов\*/

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

{

a[i]=rand()%20;

b[i]=rand()%20;

printf("a[%d] = %d\t b[%d] = %d\n",i,a[i],i,b[i]);

}

printf("\n");

/\*Сортировка массива А по возрастанию\*/

for (i = 1; i<n; i++)

{

t=a[i];

j=i-1;

while (j>=0 & a[j] >= t )

{

a[j+1]=a[j];

j=j-1;

}

a[j+1]=t;

}

/\*Сортировка массива B по убыванию\*/

for (i = 1; i<n; i++)

{

t=b[i];

j=i-1;

while (j>=0 & b[j] <= t )

{

b[j+1]=b[j];

j=j-1;

}

b[j+1]=t;

}

/\*Вывод массива\*/

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

{

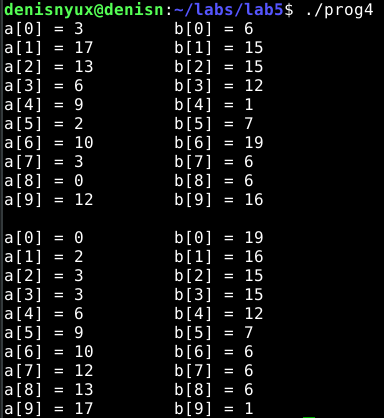
printf("a[%d] = %d\t b[%d] = %d\n",i,a[i],i,b[i]);

}

return 0;

}

**Результат выполненной работы:**

****

**Задание 5.5**

**Постановка задачи:**

Организовать ввод массива (матрицы) по столбцам.

**Математическая модель:**

Матрица вводится в двумерный массив по столбцам.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n,m | Количество элементов в строках и столбцах массива | int |
| a[n,m] | Двумерный массив | int |
| i, j | Переменная, позволяющая работать с элементами массива | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 3;

const int m = 3;

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

{

int i,j;

int a[n][m];

/\*Ввод массива\*/

printf("Enter A(3x3):\n" );

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

{

printf("\nEnter %d column:\n",i+1 );

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

{

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

}

}

/\*Вывод матрицы (для проверки)\*/

printf("\nA:\n");

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

{

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

{

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

}

printf("\n");

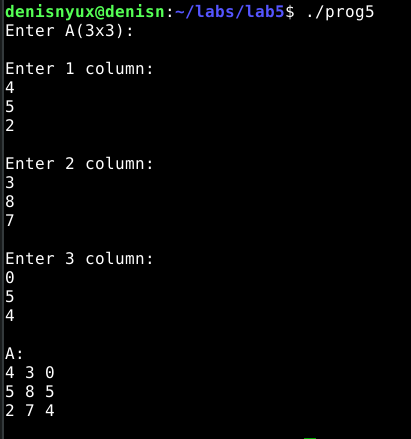
}

printf("\n");

return 0;

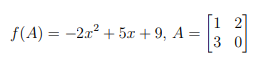
}

**Результат выполненной работы:**

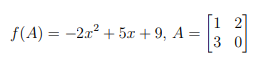
****

**Задание 5.6**

**Постановка задачи:**

Найти значения матричного многочлена f(A), где 

**Математическая модель:**

****

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n | Количество элементов в строках и столбцах массива | int |
| a[n,m] | Исходная матрица | int |
| a\_2[n,m] | Исходная матрица в квадрате | int |
| e[n,m] | Единичная матрица | int |
| x[n,m] | Значение многочлена | int |
| i, j, r | Переменные, позволяющая работать с элементами массива | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 2;

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

{

int i,j,r;

/\*A - матрица, E - единичная матрица, A\_2 - квадрат исходной матрицы, X - результат \*/

int a[2][2]={1,2,3,0},a\_2[2][2],x[2][2],e[2][2]={1,0,0,1};

/\*Вычисляем квадрат матрицы A\*/

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

{

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

{

a\_2[i][j]=0;

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

{

a\_2[i][j]=a\_2[i][j]+a[i][r]\*a[r][j];

}

}

}

/\*В цикле сразу вычисляем поэлементно матричный многочлен\*/

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

{

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

{

x[i][j]=-2\*a[i][j]+5\*a[i][j]+9\*e[i][j];

}

}

/\*Выводим результат\*/

printf("\nf(x):\n");

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

{

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

{

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

}

printf("\n");

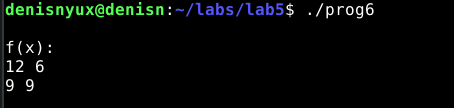
}

printf("\n");

return 0;

}

**Результат:**

****

**Задание 5.7**

**Постановка задачи:**

Транспонировать матрицу



**Математическая модель:**

Транспонированная матрица — матрица , полученная из исходной матрицы заменой строк на столбцы.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n,m | Количество элементов в строках и столбцах массива | int |
| a[2,3] | Исходная матрица | int |
| a\_t[3,2] | Транспонированная матрица | int |
| i,j | Переменные, позволяющая работать с элементами массива | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 2;

const int m = 3;

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

{

int i,j;

int a[2][3]={1,2,3,4,5,6},a\_t[3][2];

//Выводим исходную матрицу

printf("\nA\_t:\n");

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

{

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

{

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

}

printf("\n");

}

//Транспонирование

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

{

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

{

a\_t[j][i]=a[i][j];

}

}

//Вывод транспонированной матрицы

printf("\nA\_t:\n");

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

{

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

{

printf("%d ",a\_t[i][j]);

}

printf("\n");

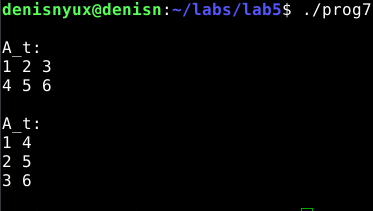
}

printf("\n");

return 0;

}

**Результат:**

****

**Задание 5.8**

**Постановка задачи:**

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



**Математическая модель:**

При умножении матриц стоит учитывать размерность. При умножении A на AT получится матрица размерностью 2х2, а при умножении АТ на А, 4х4.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n,m | Количество элементов в строках и столбцах массива | int |
| a[2,4] | Исходная матрица | int |
| a\_t[4,2] | Транспонированная матрица | int |
| с[2][2] | Произведение матрицы на транспонированную | int |
| x[4][4] | Произведение транспонированной матрицы на исходную | int |
| r | Вспомогательная переменная при умножении матриц | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n=2;

const int m=4;

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

{

int i,j,r;

int a[2][4]={1,2,1,3,4,-1,5,-1},a\_t[4][2],c[2][2],x[4][4];

//Транспонировани

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

{

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

{

a\_t[j][i]=a[i][j];

}

}

/\*Умножение матрицы А на матрицу A\_t\*/

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

{

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

{

c[i][j]=0;

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

{

c[i][j]+=a[i][r]\*a\_t[r][j];

}

}

}

/\*Умножение матрицы A\_t на А\*/

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

{

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

{

x[i][j]=0;

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

{

x[i][j]+=a\_t[i][r]\*a[r][j];

}

}

}

printf("\nA \* A\_t:\n");

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

{

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

{

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

}

printf("\n");

}

printf("\nA\_t \* A:\n");

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

{

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

{

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

}

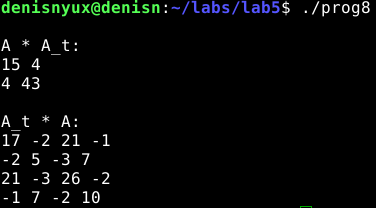
printf("\n");

}

return 0;

}

**Результат выполненной работы:**



**Задание 5.9**

**Постановка задачи:**

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



**Математическая модель:**

Произведение данных матриц должно в обоих случаях быть равным



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n,m | Количество элементов в строках и столбцах массива | int |
| a[2,2], b[2][2] ,c[2][2] | Исходные матрица | int |
| ab[2][2] | Произведение A на B | int |
| ab\_с[2][2] | (A\*B)\*C | int |
| bc[2][2] | Произведение В на С | int |
| a\_bc[2][2] | A\*(B\*C) | int |
| r | Вспомогательная переменная при умножении матриц | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n=2;

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

{

int i,j,r,t;

int a[2][2]={1,-1,-1,1},b[2][2]={2,0,-3,1},c[2][2]={3,-1,2,3};

int ab[2][2],ab\_c[2][2],bc[2][2],a\_bc[2][2];

/\*Находим первое произведение A\*B \*/

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

{

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

{

ab[i][j]=0;

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

{

ab[i][j]+=a[i][r]\*b[r][j];

}

}

}

/\*Находим второе произведение (A\*B)\*С \*/

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

{

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

{

ab\_c[i][j]=0;

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

{

ab\_c[i][j]+=ab[i][r]\*c[r][j];

}

}

}

/\*Находим первое произведение B\*C \*/

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

{

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

{

bc[i][j]=0;

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

{

bc[i][j]+=b[i][r]\*c[r][j];

}

}

}

/\*Находим второе произведение A\*(B\*С) \*/

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

{

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

{

a\_bc[i][j]=0;

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

{

a\_bc[i][j]+=a[i][r]\*bc[r][j];

}

}

}

/\*Выводи матрицы\*/

printf("\n(A\*B)\*C:\n");

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

{

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

{

printf("%d ",ab\_c[i][j]);

}

printf("\n");

}

printf("\n(A\*B)\*C:\n");

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

{

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

{

printf("%d ",a\_bc[i][j]);

}

printf("\n");

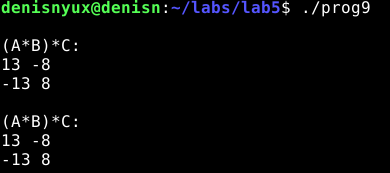
}

printf("\n");

return 0;

}

**Результат выполненной работы:**

****

**Задача 5.10**

**Постановка задачи:**

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

**Математическая модель:**

Построчно, во вложенном цикле будем считать сначала среднее арифметическое, а затем во втором вложенном цикле будем заменять элементы, не забыв при этом после каждого перехода на следующую строчку обнулять значение среднего арифметического.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| **Имя идентификатора** | **Семантика** | **Тип** |
| n | Количество элементов в строках и столбцах массива | int |
| a[n,n] | Исходная матрица | int |
| t | Среднее арифметическое | int |
| i,j | Переменные, позволяющая работать с элементами массива | int |

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

const int n = 5;

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

{

int i,j,t=0;

int a[n][n];

/\*Заполнение массива случайными числами\*/

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

{

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

{

a[i][j]=rand()%20;

}

}

printf("\nA:\n");

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

{

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

{

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

}

printf("\n");

}

printf("\n");

/\*Нахождение общего арифметического и замена элементов\*/

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

{

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

{

t+=a[i][j];

}

t=t/n;

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

{

a[i][j]=t;

}

t=0;

}

/\*Вывод полученного массива\*/

printf("\nA:\n");

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

{

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

{

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

}

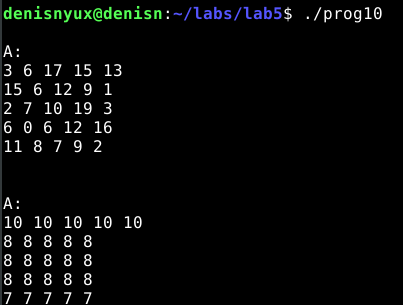
printf("\n");

}

return 0;

}

**Результат выполненной работы:**

****