**Лабораторная работа №5**

**Тема лабораторной работы**

**Задача 1**

**Постановка задачи**

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

**Математическая модель**

y = x \* x

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| x | вводимый с клавиатуры массив | int |
| y | результат в виде массива | int |
| i | параметр цикла | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void)

{

int i,x[5],y[5];

printf("x[5]\n");

for (i =1;i<=5;i++)

{

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

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

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

}

printf("y[5] \n");

for (i = 1;i<=5;i++)

{

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

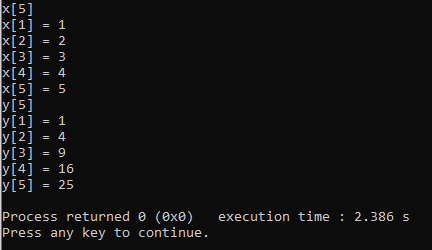
printf("%d\n", y[i]);

}

return 0;

}

**Результат выполненной работы**



**Задача 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**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | промежуточная переменная | int |
| i | параметр цикла |
| x[i] | массив |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void)

{

int i,x[5],a;

printf("x[5]\n");

for (i =1;i<=5;i++)

{

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

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

}

printf("y[5]\n");

for (i =1;i<=5/2;i++)

{

a = x[i];

x[i] = x[6-i];

x[6-i] = a;

}

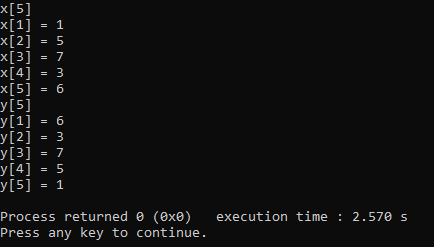
for (i=1;i<=5;i++)

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

return 0;

}

**Результат выполненной работы**



**Задача 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**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | промежуточная переменная | int |
| i | параметр цикла | int |
| k | параметр цикла | int |
| x[i] | массив | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void)

{

int a,i,f,x[10],k;

for (i = 0; i < 10; i++) {

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

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

}

printf("y[10]\n");

for(k = 0; k<9; k++)

{

for (i = 0;i<10-k; i++){

if (x[i]>x[i+1]){

a = x[i];

x[i] = x[i+1];

x[i+1] = a;

}}}

for (i = 0; i < 10; i++) {

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

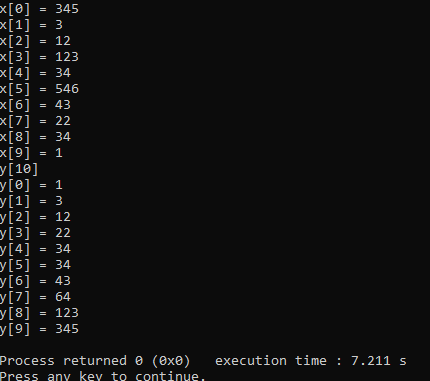
printf("%d\n",x[i]);

}

return 0;

}

**Результат выполненной работы**



**Задача 4**

**Постановка задачи**

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

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | промежуточная переменная | int |
| i | параметр цикла | int |
| x[i] | массив | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void) {

int a,i,x[10];

for (i = 0; i < 10; i++) {

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

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

}

for(int k = 1; k<10; ++k)

{

a = x[k];

for (i = k-1;i>=0 && x[i]> a; i--){

x[i + 1]= x[i];

x[i] = a;

}

}

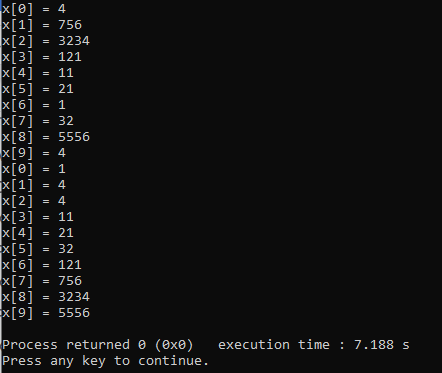
for (i = 0; i < 10; i++)

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

return 0;

}

**Результат выполненной работы**



**Задача 5**

**Постановка задачи**

Организовать ввод массива (матрицы) по столбцам.

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j | параметр цикла | int |
| x[i][j] | двумерный массив | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void)

{

int i,j,x[2][4];

for (i = 0;i<4;i++)

{

for (j = 0; j<2; j++)

{

printf ("x[%d][%d] = ", j+1,i+1 );

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

}

printf("\n");

}

for (j = 0;j<2;j++)

{

for (i = 0; i<4; i++)

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

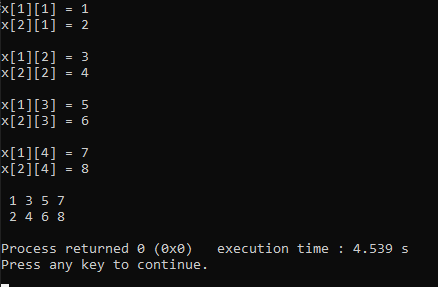
printf("\n");

}

return 0;

}

**Результат выполненной работы**

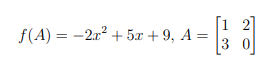


**Задача 6**

**Постановка задачи**

Найти значения матричного многочлена f(A)

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j | параметр цикла | int |
| b[i][j] | двумерный массив | int |
| a[i][j] | двумерный массив | int |
| c[i][j] | двумерный массив | int |
| k | параметр цикла | int |
| e | элемент единичной матрицы | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void) {

int k,i,j,e,b[2][2] = {0},a[2][2] = {1,2,3,0},c[2][2];

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

{

for(j = 0;j<2;j++)

{

for ( k = 0; k< 2; k++)

{

b[i][j] = b[i][j] + a[i][k]\*a[k][j];

}

if (i == j)

e = 1;

else

e = 0;

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

}

}

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

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

{

for(j = 0;j<2;j++)

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

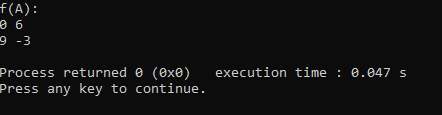
printf("\n");

}

return 0;

}

**Результат выполненной работы**



**Задача 7**

**Постановка задачи**

Транспонировать матрицу

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j | параметр цикла | int |
| a[i][j] | двумерный массив | int |
| b[j][i] | двумерный массив | int |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void)

{

int i,j,a[2][3]={1,2,3,4,5,6},b[3][2];

for (i = 0;i<3;i++)

{

for (j = 0;j<2;j++)

{

b[i][j]=a[j][i];

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

}

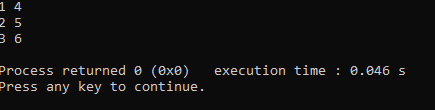
printf("\n");

}

return 0;

}

**Результат выполненной работы**



**Задача 8**

**Постановка задачи**

Вычислить произведения A × AT и AT × A

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j |
| k |
| b[i][j] | двумерный массив |
| a[j][i] |
| d[j][j] |
| c[i][i] |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void) {

int i,j,k,b[4][2]= {0},a[2][4] = {1,2,1,3,4,-1,5,-1},c[4][4]={0},d[2][2]={0};

printf("A^T: \n");

for(i = 0;i<4;i++)

{

for(j = 0;j<2;j++)

{

b[i][j] = a[j][i];

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

}

printf("\n");

}

printf("A\*A^T\n");

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

{

for(j = 0;j<2;j++)

{

for ( k = 0; k<4; k++)

d[i][j] = d[i][j] + a[i][k]\*b[k][j];

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

}

printf("\n");

}

printf("A^T\*A\n");

for(i = 0;i<4;i++)

{

for(j = 0;j<4;j++)

{

for ( k = 0; k<2; k++)

c[i][j] = c[i][j] + a[k][i]\*b[j][k];

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

}

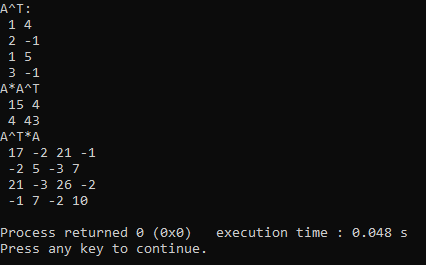
printf("\n");

}

return 0;

}

**Результат выполненной работы**



**Задача 9**

**Постановка задачи**

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

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j |
| k |
| b[2][2] | двумерный массив |
| a[2][2] |
| c[2][2] |
| ab[2][2] |
| abc | результат |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void) {

int i,j,k,b[2][2]= {2,0,-3,1},a[2][2] = {1,-1,-1,1},c[2][2]={3,-1,2,3},ab[2][2]={0},abc;

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

for(j = 0;j<2;j++)

for ( k = 0; k<2; k++)

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

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

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

{

for(j = 0;j<2;j++)

{

abc=0;

for ( k = 0; k<2; k++)

abc = abc + ab[k][i]\*c[j][k];

printf(" %d",abc);

}

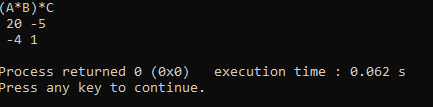
printf("\n");

}

return 0;

}

**Результат выполненной работы**



**Задача 10**

**Постановка задачи**

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

**Математическая модель**

k = k+a[i][j]

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| j | параметр цикла | int |
| k | накопительная переменная | float |
| a[i][j] | двумерный массив | float |

**Код программы**

#include <stdio.h>

#include <math.h>

int main(void) {

int i,j;

float k,a[3][3] = {1,5,5,4,0,3,9,9,1};

for(i = 0;i<3;i++)

{

k=0;

for(j = 0;j<3;j++)

{

k = k+a[i][j];

}

k=k/3;

a[i][0]=k;

}

for(i = 0;i<3;i++)

{

for (j = 0;j<3;j++)

printf(" %.1f",a[i][j]);

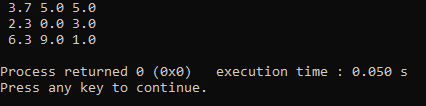
printf(" \n");

}

return 0;

}

**Результат выполненной работы**



**Задача 11**

**Постановка задачи**

Создать динамический одномерный массив целых чисел F размерности 12 и заполнить его положительными и отрицательными числами, где R — сумма отрицательных элементов F, Q — отрицательный элемент массива, S — произведение положительных элементов массива F.

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| \*f | указатель | int |
| \*b | указатель | int |
| k | длина массива | int |
| j | накопительная переменная | int |
| n | длина массива | int |
| t | результат | float |
| r | накопительная переменная | float |
| s | накопительная переменная | float |
| q | промежуточная переменная | float |

**Код программы**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

int main(void) {

int i,\*f, \*b, k, j,n=12;

float t,r,s,q ;

r = 0;

s = 1;

f = (int\*)calloc(n, sizeof(int));

b = (int\*)malloc(k\*sizeof(int));

j = 0;

if (f == NULL || b == NULL)

return 0;

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

{

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

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

if (f[i]<0)

{

r= r+f[i];

b[j] = f[i];

j = j+1;

}

if(f[i]>0)

s = s\*f[i];

}

k = j;

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

{q = b[i];

t = (r+q+s)/(r\*q\*s+2);

printf(" T%d = ",i);

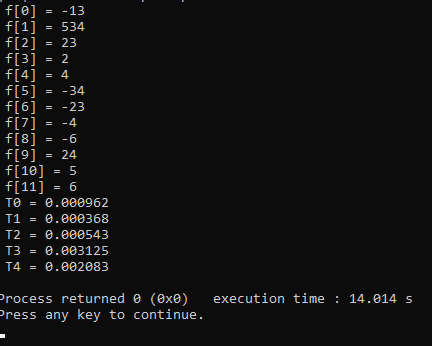
printf("%f\n", t );

}

return 0;

}

**Результат выполненной работы**



**Задача 12**

**Постановка задачи**

Дан одномерный массив целых чисел A размера 12. Вычислить: где S — количество элементов массива A с нечётными индексами, T — наибольший по модулю элемент массива A, U — сумма отрицательных элементов массива A.

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| y | результат | float |
| u | накопительная переменная |
| t | промежуточная переменная |
| s | накопительная переменная |
| n | длина массива | int |
| i | параметр цикла |
| \*a | указатель |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(void) {

int \*a,i,n=12;

float y,t = 0 ,s = 0, u = 0;

a = (int\*)calloc(n,sizeof(int));

printf("A[12]\n");

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

{

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

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

if (i%2 != 0)

s = s+1;

if (a[i]<0)

u = u+1;

if (t<abs(a[i]))

t = a[i];

}

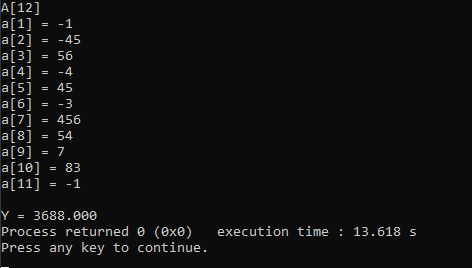
y = (u+t)\*(s+2);

printf("\nY = %.3f", y);

return 0;

}

**Результат выполненной работы**



**Задача 13**

**Постановка задачи**

Напишите программу для вычисления пересечения двух конечных множеств (наборов) A и B целых чисел одинакового размера с использованием динамических массивов. В качестве множества A можно взять первые 12 чисел ряда Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144. В качестве множества B можно взять первые 12 чисел последовательности Падована: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16. Дублирующиеся значения можно исключать.

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| i | параметр цикла | int |
| \*a | указатель |
| \*b | указатель |
| k | длина массива |
| j | параметр цикла |
| n | длина массива |
| \*c | указатель |
| s | накопительная переменная |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(void) {

int j,\*a = NULL,\*b = NULL,\*c = NULL,i,s, k, n =12;

a = (int\*)calloc(n,sizeof(int));

b = (int\*)calloc(n,sizeof(int));

c = (int\*)malloc(k\*sizeof(int));

if (a == NULL || b== NULL || c == NULL)

{

printf("Error\n");

}

s=1;

printf("Fibonacci series:");

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

{ if (i<2)

a[i] = s;

else

a[i] = a[i-2]+a[i-s];

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

}

printf("\nPadovan sequence:");

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

{

if (i<3)

b[i] = s;

else

b[i] = b[i-2]+b[i-3];

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

}

s = 0;

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

{

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

{

if (a[i] == b[j] && a[i]!= a[i+1] && b[j]!= b[j+1])

{

c[s]= a[i];

s = s + 1;

}

k = s;

}}

printf("\n");

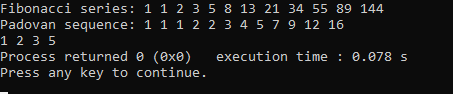
for (i = 0 ;i<k;i++)

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

return 0;

}

**Результат выполненной работы**



**Задача 14**

**Постановка задачи**

Выделить динамически память под некоторую матрицу A размерности M × N и заполнить её произвольными числами. Сократить правильно размер этой матрицы, удалив из неё одну выбранную строку, освободив от неё также и память, используя указатели. После удаления строки в матрице AM×N должна быть возможность обхода всех элементов «новой» матрицы AM−1×N таким же способом, что и изначальной матрицы. Распечатать (используя циклы) матрицу до удаления строки и после удаления.

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| n | столбцы | int |
| m | строки |
| i | параметр цикла |
| j | параметр цикла |
| \*\*a | указатель |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int n,m,i,j,\*\*a = NULL;

printf("M:");

scanf("%d", &m);

printf("N:");

scanf("%d",&n);

printf("Source array:\n");

a = (int\*\*)malloc(m\*sizeof(int\*));

if (a == NULL)

{

printf("Error");

return 0;

}

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

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

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

{

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

{

a[i][j] = rand() % 100;

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

}

printf("\n");

}

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

a[i-1][j]=NULL;

m -=1;

printf("New matrix:\n");

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

{

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

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

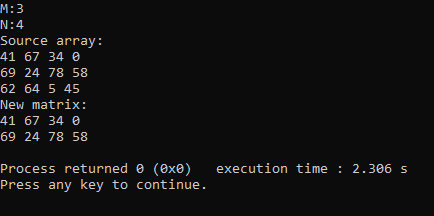
printf("\n");

}

return 0;

}

**Результат выполненной работы**



**Задача 15**

**Постановка задачи**

Написать программу, которая вычисляет некоторый вектор b, как результат умножения некоторой матрицы M на вектор a:

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| \*\*a | указатель | int |
| \*b | указатель |
| \*c | указатель |
| m | строка |
| n | столбец |
| i | параметр цикла |
| j | параметр цикла |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int \*\*a=NULL, \*b = NULL,\*c = NULL,m,n,i,j;

printf("m:");

scanf("%d", &m);

printf("n:");

scanf("%d",&n);

a = (int\*\*)malloc(m\*sizeof(int\*));

b = (int\*)malloc(n\*sizeof(int\*));

c = (int\*)malloc(m\*sizeof(int\*));

if (a == NULL|| b == NULL || c==NULL)

{

printf("Error");

return 0;

}

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

a[i] = (int\*)malloc(m\*sizeof(int));

printf("M[m][n]:\n");

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

{

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

{

a[i][j] = rand() % 100;

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

}

printf("\n");

}

printf("a:\n");

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

{

b[i] = rand() % 100;

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

}

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

{

c[i] = 0;

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

{

c[i] = c[i]+ a[i][j]\*b[j];

}

}

printf("Resulting vector:\n");

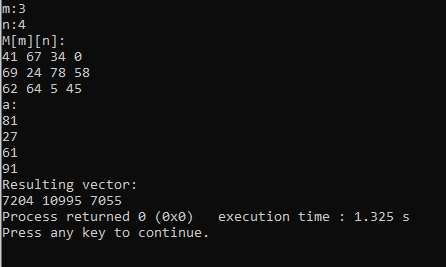
for (i = 0;i<m;i++)

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

return 0;

}

**Результат выполненной работы**



**Задача 16**

**Постановка задачи**

Напишите программу, в которой создаётся квадратная матрица, заполненная нулями и единицами. Единичные значения у тех элементов, для которых сумма индексов является нечётным числом. Нулевые значения у тех элементов, для которых сумма индексов является чётным числом.

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| n | размер массива | int |
| i | параметр цикла |
| j | параметр цикла |
| \*\*a | указатель |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int \*\*a=NULL,n,i,j;

printf("Size of the square matrix: ");

scanf("%d",&n);

a = (int\*\*)malloc(n\*sizeof(int\*));

if (a == NULL)

{

printf("Error");

return 0;

}

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

{

a[i] = (int\*)malloc(n\*sizeof(int));

}

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

{

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

{

if ((j+i)%2!= 0)

a[i][j] = 1;

else

a[i][j] = 0;

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

}

printf("\n");

}

return 0;

}

**Результат выполненной работы**

