САА - Lab 2

Problem 12\*

Design an algorithm that finds the product of only those elements of a one-dimensional array that are part of pairs whose sum of elements is not greater than 12.

Problem 13\*

Design an algorithm that finds the number of blocks in a one-dimensional array. A block is a sequence of two or more identical elements.

VI. Algorithms with 2D arrays

Problem 14

Inputting and outputting the elements of a two-dimensional array.

#include <iostream>

using namespace std;

#define N 2

#define M 2

int main()

{

int i, j, arr[N][M];

cout << "Enter the elements of two-dimensional array\n";

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

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

{

cout << "arr[" << i << "][" << j << "] = ";

cin >> arr[i][j];

}

cout << "The elements of two-dimensional array\n";

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

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

cout<<"arr["<<i<<"]["<<j<<"] = "<<arr[i][j]<<"\n";

return 0;

}

Problem 15\*

Design an algorithm that finds the sum of the elements along the perimeter of a two-dimensional array.

Problem 16\*

Using a two-dimensional array find the sum of the elements on the main diagonal, the sums of the elements of each row, and the number of elements below the main diagonal with values that are smaller than the sum of their indices. Put all of these results in a one-dimensional array.

Problem 17

Finding the elements of a two-dimensional array with values in the interval [x, y].

#include <iostream>

using namespace std;

#define N 2

#define M 3

int main()

{

int i, j, arr1[N][M], x=0, y=0, k=0, q=0;

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

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

{

cout << "arr1[" << i << "][" << j << "] = ";

cin >> arr1[i][j];

}

cout << "Enter x and y\n";

cout << "x = ";

cin >> x;

cout << "y = ";

cin >> y;

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

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

if(arr1[i][j] >= x && arr1[i][j] <= y)

k++;

int \*p = new int[k];

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

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

if(arr1[i][j] >= x && arr1[i][j] <= y)

{

\*(p+q) = arr1[i][j];

q++;

}

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

cout << "arr2[" << i << "] = " << \*(p+i) << "\n";

return 0;

}

Problem 18

Write a program that processes the elements of a two-dimensional array as follows:   
a) all odd elements above the main diagonal become even (by adding 1);   
b) all even elements below the main diagonal become odd (by subtracting 1);   
c) for each change done according to a and b, the element of the main diagonal located on the same row as the change should increase by 1.

#include <iostream>

using namespace std;

#define N 2

#define M 2

int main()

{

int i, j, arr[N][M];

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

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

{

cout << "arr[" << i << "][" << j << "] = ";

cin >> arr[i][j];

}

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

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

{

if((j>i) && (arr[i][j]%2!=0))

{

arr[i][j] = arr[i][j] + 1;

arr[i][i]++;

}

if((i>j) && (arr[i][j]%2==0))

{

arr[i][j] = arr[i][j] - 1;

arr[i][i]++;

}

}

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

{

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

cout << arr[i][j] << " ";;

cout << "\n";

}

return 0;

}

Problem 19\*

Design an algorithm that finds the sum of the elements on the secondary diagonal of a two-dimensional array.

Problem 20

Using a unique index to display the corresponding element of a two-dimensional array.

#include <iostream>

#define N 2

#define M 2

using namespace std;

int main()

{

int i, j, arr[N][M], \*p, n;

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

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

{

cout << "arr[" << i << "][" << j << "] = ";

cin >> arr[i][j];

}

p = &arr[0][0];

cout << "n: ";

cin >> n;

cout << "arr[" << n << "] = " << \*(p+n-1) << "\n";

return 0;

}

1. Run and analyze the solved problems – 14, 17, 18 and 20

2. Design algorithms and write programs that solve the remaining problems – 12\*, 13\*, 15\*, 16\* and 19\*