Тема 21. Работа с массивами С++. Адресная арифметика.

Селиванова Вера

Декабрь 2016

1 Задача

В матрице A(4,4), содержащей вещественные элементы, в каждом столбце поменять местами максимальный элемент с диагональным. Распечатать:

а) исходную и преобразованную матрицы;

б) адреса и значения тех элементов, которые оказались максимальными.

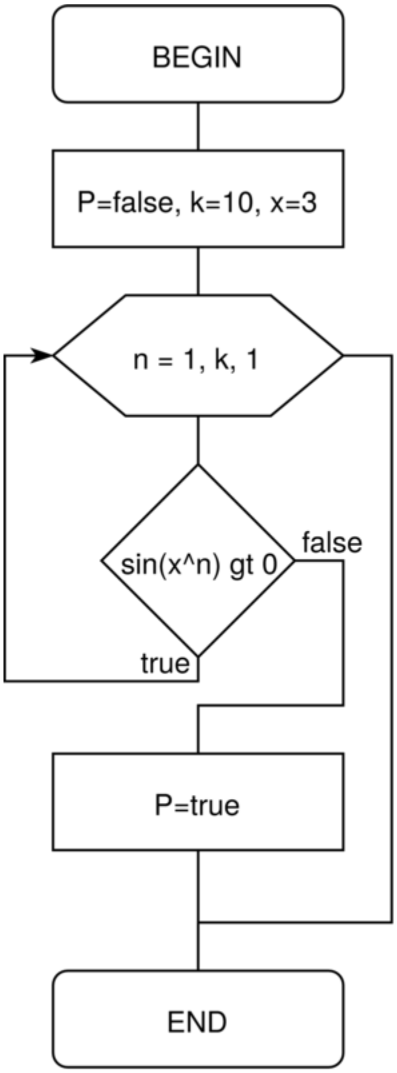
2 Программный код

```
#include <iostream>

using namespace std;

int main(void) {

  // Matrix dimentions
  const unsigned int DIMENSION = 4;

  // i, j - The counter cycles' variables
  // The p - is a temporary variable contains position of
  //         the biggest value in iterating column.
  unsigned int i, j, p;

  // The biggest value is found in column on iteration.
  // The tmp - is a temporary variable helps to swap the
  //           biggest value in column and diagonal value.
  float biggest, tmp;

  // The matrix
  float matrix[DIMENSION][DIMENSION] = {
    {32, 12, 65, 23},
    {61, 31, 94, 71},
    {45, 37, 82, 67},
    {54, 86, 63, 10}
  };

  cout << endl << "The␣origin␣matrix:" << endl;

  // Show user the origin matrix that we will be changed in
  // accordance with the exercise conditions.
  for (i = 0; i < DIMENSION; i++) {
    for (j = 0; j < DIMENSION; j++)
      cout << matrix[i][j] << "␣";
    cout << endl;
  }

  cout << endl;

  for (i = 0; i < DIMENSION; i++) {
    // Resetting the biggest variable before iterating over
    // the next one column.
    biggest = 0;
    // Iterating over another one column to find the biggest
    // element in it.
    for (j = 0; j < DIMENSION; j++)
      if (matrix[j][i] > biggest) {
        biggest = matrix[j][i];

        // Memorizing position of the biggest element. It
        // help us to swap the biggest value with a diagonal
        // element.
        p = j;
      }

    cout << "Column␣" << i << ".␣The␣biggest␣value␣is␣" <<
      biggest << ",␣found␣on␣" << p << "␣place." <<
      "\nIt␣should␣be␣swapped␣with␣diagonal␣element␣on␣" <<
      4 - i - 1 << "␣place." << endl;
    cout << endl;

    // Swapping biggest element with element in column
    // expected to be diagonal in the matrix.
    tmp = matrix[DIMENSION - 1 - i][i];
    matrix[DIMENSION - 1 - i][i] = matrix[p][i];
    matrix[p][i] = tmp;
  }

  cout << endl << "The␣result␣matrix:" << endl;

  // Show user the result matrix, in which the biggest
  // elements in its columns were swapped with elements
  // layed on the main matrix diagonal.
  for (i = 0; i < DIMENSION; i++) {
    for (j = 0; j < DIMENSION; j++)
      cout << matrix[i][j] << "␣";
    cout << endl;
  }

  return 0;
}
```

3 Результат работы

```
The origin matrix:
32 12 65 23
61 31 94 71
45 37 82 67
54 86 63 10

Column 0. The biggest value is 61, found on 1 place.
It should be swapped with diagonal element on 3 place.

Column 1. The biggest value is 86, found on 3 place.
It should be swapped with diagonal element on 2 place.

Column 2. The biggest value is 94, found on 1 place.
It should be swapped with diagonal element on 1 place.

Column 3. The biggest value is 71, found on 1 place.
It should be swapped with diagonal element on 0 place.


The result matrix:
32 12 65 71
54 31 94 23
45 86 82 67
61 37 63 10
```

4 Блок схема


BEGIN
P=false, k=10, x=3
n = 1, k, 1
sin(x^n) gt 0
false
true
P=true
END