Міністерство освіти і науки України

Національний Технічний Університет

«Харківський Політехнічний Інститут»

Кафедра «Стратегічного Керування»

Лабораторная робота №4

«Використання масивів і вказівників»

Перевірила: ас. кафедри СК

Вільхівская О. В.

Виконав:

Харків, 2017

Лабораторная работа №4

Тема работы: Разработка **программ с применением** функций

Задания

1. Сума мінімального і максимального елементів

Написати програму, яка обчислює суму мінімального і максимального елементів масиву значень з плаваючою точкою подвійної точності. Здійснити пошук мінімального і максимального елементів у двох окремих функціях.

1. Сортування за зменшенням

Написати програму, яка сортує елементи масиву цілих чисел за зменшенням.

1. Сума додатних елементів

Написати програму, яка обчислює суму додатних елементів двовимірного масиву.

1. Масив у динамічній пам'яті

Написати програму, яка зчитує з клавіатури кількість рядків і стовпців двовимірного масиву, розташовує масив у динамічній пам'яті, зчитує елементи масиву з клавіатури, обчислює суми рядків і записує ці суми в новий масив.

1. Індивідуальне завдання

Створити програму, яка визначає та ініціалізує двовимірний масив цілих елементів, а потім реалізує такі дії:

* перетворення вихідного масиву відповідно до завдання, наведеного в колонці "Перший крок"
* створення в динамічній пам'яті та заповнення одновимірного масиву чисел типу double відповідно до завдання, наведеного в колонці "Другий крок"
* виведення на екран елементів обох масивів і звільнення пам'яті

Треба передбачити виведення повідомлень про помилки, якщо перетворення або заповнення неможливі.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер варіанту  (номер студента у списку)** | **Перший крок.  правило перетворення елементів першого масиву:** | **Другий крок.  правило заповнення елементів другого масиву:** | **Кількість рядків m** | **Кількість стовпців n** |
| 2 | Усі елементи з парними значеннями повинні бути замінені їх квадратами | Кубічні корені мінімальних елементів колонок | 3 | 5 |

Хід роботи

1. Сума мінімального і максимального елементів

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  using namespace std;  double max(double a[]) {  double max = a[0];  for (int i = 1; i < 4; i++) {  if (a[i] > max) {  max = a[i];  }  }  return max;  }  double min(double a[]) {  double min = a[0];  for (int i = 1; i < 4; i++) {  if (a[i] < min) {  min = a[i];  }  }  return min;  }  void main()  {  const int n = 4;  double a[] = { -1, -2, 44.4534, 8 };  double sum = max(a) + min(a);  cout << sum << endl;  system("pause");  } |

Пример работы программы:

|  |
| --- |
| 42.4534  Для продолжения нажмите любую клавишу . . . |

1. Сортування за зменшенням

|  |
| --- |
| #include "stdafx.h"  #include<iostream>  using namespace std;  void main()  {  const int n = 7;  double a[] = {-1, -2, 3, -3, -7, 9, 0 };  bool mustSort; // повторюємо сортування  // якщо mustSort дорівнює true  do  {  mustSort = false;  for (int i = 0; i < n-1; i++)  {  if (a[i] < a[i + 1])  // Обмінюємо елементи  {  double temp = a[i];  a[i] = a[i + 1];  a[i + 1] = temp;  mustSort = true;  }  }  } while (mustSort);  for (int i = 0; i < n; i++)  {  cout << a[i] << ' ';  }  system("pause");  } |

Пример выполнения программы:

|  |
| --- |
| 9 3 0 -1 -2 -3 -7 Для продолжения нажмите любую клавишу . . . |

1. Сума додатних елементів

|  |
| --- |
| #include "stdafx.h"  #include<iostream>  using namespace std;  void main()  {  int a[][3] = { { -1,2,3 },  { 2,-3,4 },  { 0,-1,3 },  { 1,-1,12 } };  int sum = 0;  for (int i = 0; i < 4; i++)  {  for (int j = 0; j < 3; j++)  {  if (a[i][j]>0) {  sum += a[i][j];  }  }  }  cout << sum << endl;  system("pause");  } |

Пример выполнения программы:

|  |
| --- |
| 27  Для продолжения нажмите любую клавишу . . . |

1. Масив у динамічній пам'яті

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  using namespace std;  void main() {  int m, n;  int i;  cout << "Rows and Columns" << endl;  cin >> m >> n;  int \*\*arr = new int\*[m];  for (i = 0; i < m; i++) {  arr[i] = new int[n];  }  int \*arr2 = new int[m];  cout << "Elements" << endl;  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  cin >> arr[i][j];  }  }  for (int i = 0; i < m; i++) {  int sum = 0;  for (int j = 0; j < n; j++) {  sum += arr[i][j];  }  arr2[i] = sum;  }  cout << "Sums of the Rows:" << endl;  for (int i = 0; i < m; i++) {  cout << arr2[i] << ' ';  }  for (i = 0; i < m; i++)  {  delete[] arr[i];  }  delete[] arr;  delete [] arr2;  system("pause");  } |

Пример выполнения программы:

|  |
| --- |
| Rows and Columns  3 2  Elements  1 4  6 7  -9 1  Sums of the Rows:  5 13 -8 Для продолжения нажмите любую клавишу . . . |

1. Індивідуальне завдання

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  using namespace std;  void main()  {  int i = 0;  int j = 0;  const int m = 3, n = 5;  int arr[m][n] = { {1,2,3,4,5},  {6,7,8,9,10},  {9,8,7,6,5} };  int arr1[m][n];  cout << "Initial Array: " << endl;  for (i = 0; i < m; i++) {  for (j = 0; j < n; j++) {  cout << arr[i][j] << ' ';  }  cout << endl;  }  for (i = 0; i < m; i++) {  for (j = 0; j < n; j++) {  (i + j) % 2 == 0 ? arr1[i][j] = arr[i][j]\*arr[i][j] : arr1[i][j]= arr[i][j];  }  }  cout << endl << "Array 1:" << endl;  for (i = 0; i < m; i++) {  for (j = 0; j < n; j++) {  cout << arr1[i][j] << ' ';  }  cout << endl;  }  double \*p = new double[n];  double temp;  for (j = 0; j < n; j++) {  temp = arr[0][j];  for (i = 1; i < m; i++) {  if (arr[i][j]<temp) {  temp = arr[i][j];  }  }  p[j] = pow(temp,(double)1/3);  }  cout << endl << "Array 2:" << endl;  for (i = 0; i < n; i++) {  cout << p[i] << ' ';  }  delete p;  system("pause");  } |

Пример выполнения программы:

|  |
| --- |
| Initial Array:  1 2 3 4 5  6 7 8 9 10  9 8 7 6 5  Array 1:  1 2 9 4 25  6 49 8 81 10  81 8 49 6 25  Array 2:  1 1.25992 1.44225 1.5874 1.70998 Для продолжения нажмите любую клавишу . . . |

Висновок

C++ дозволяє нам створювати та оперувати масивами хоч у роботі з ними є багато обмежень. Щоб задати масив програміст повинен знати його розмір, або кількість стовпчиків (якщо це двовимірний масив) ще на самому початку. Обійти це обмеження дозволяють вказівники – змінні, що містять адресу пам’яті. Операція ж new дозволяє розмістити у пам’яті цілий масив, що є дуже корисним при створенні динамічних масивів.