Міністерство освіти і науки України

Національний Технічний Університет

«Харківський Політехнічний Інститут»

Кафедра «Стратегічного Керування»

Лабораторна робота № 5

«Робота зі структурами та файлами»

Перевірила: ас. кафедри СК

Вільхівская О. В.

Виконав:

Харків, 2017

Лабораторна робота №

Тема:

Завдання

1. Точки у тривимірному просторі

Точки у тривимірному просторі

1. Середнє арифметичне

Середнє арифметичне

1. Індивідуальне завдання

Написати програму, яка забезпечує файлове введення та виведення і включає індивідуальне завдання попередньої лабораторної роботи. Слід реалізувати такі дії:

* визначення константи (n) яка визначає кількість стовпців двовимірного масиву
* відкриття файлу для читання (файл повинен бути підготовлений за допомогою текстового редактора)
* читання цілих чисел до кінця файлу і зберігання їх у зв'язаному списку
* створення двовимірного масиву в динамічній пам'яті; кількість рядків повинна бути обчислена на основі кількості зчитаних з файлу значень та визначеної стовпців
* заповнення двовимірного масиву рядок за рядком; відсутні елементи останнього рядка повинні бути заповнені нулями
* видалення елементів зв'язаного списку з динамічної пам'яті
* реалізація попереднього індивідуального завдання
* зберігання результатів в новому файлі
* видалення масивів операцією delete.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. Точки у тривимірному просторі

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  #include <fstream>  using namespace std;  struct point {  double x,y,z;  };  double sqr(double x) {  return x\*x;  }  double distance(point p1, point p2) {  return std::sqrt(sqr(p1.x + p2.x) + sqr(p1.y + p2.y) + sqr(p1.z + p2.z));  }  void main()  {  point p1 = { 0, 0, 0 };  point p2 = { 3, 4, 0 };  cout << distance(p1, p2);  system("pause");  } |

Приклад виконання програми:

|  |
| --- |
| 5Для продолжения нажмите любую клавишу . . . |

1. Середнє арифметичне

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  #include <fstream>  using namespace std;  void main()  {  ifstream in("Data.txt");  double x, count = 0, sum = 0;  while (in >> x) {  count += 1;  sum += x;  }  printf("%f", sum/count);  system("pause");  } |

Файл Data.txt у каталозі програми

|  |
| --- |
| 1 4  5 2 |

Приклад виконання програми:

|  |
| --- |
| 3.000000Для продолжения нажмите любую клавишу . . . |

1. Індивідуальне завдання

|  |
| --- |
| #include "stdafx.h"  #include <iostream>  #include <fstream>  using namespace std;  struct list {  double data;  list\* next;  };  double \*\*readFromFile(char \*fileName, int &count, int n, int &m) {  list\* first = 0;  list\* last = 0;  list\* link = 0;  ifstream in(fileName);  double d;  while (in >> d) {  count++;  link = new list;  link->data = d;  link->next = 0;  if (last == 0) {  first = last = link;  }  else {  last->next = link;  }  last = link;  }  m = count / n;  if (count%n != 0) {  int var = n - (count - m\*n);  for (int i = 0; i < var; i++) {  count++;  link = new list;  link->next = 0;  link->data = 0;  last->next=link;  last = link;  }  }  ///////////////////////////////////////  m = count / n;  double \*\*arr = new double\*[m];  link = first;  for (int i = 0; i < m; i++) {  arr[i] = new double[n];  for (int j = 0; j < n; j++) {  arr[i][j] = link->data;  link = link->next;  }  }  ///////////////////////////////////////  while (first) {  link = first;  first = first->next;  delete link;  }  return arr;  }  void initial(char \*filename, double \*\*arr, int m, int n)  {  ofstream out(filename);  out << "Initial Array:" << endl;  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  out << arr[i][j] << ' ';  }  out << endl;  }  }  void root(char \*fileName, double \*\*arr, int m, int n) {  double \*p = new double[n];  double temp;  for (int j = 0; j < n; j++) {  temp = arr[0][j];  for (int i = 1; i < m; i++) {  if (arr[i][j]<temp) {  temp = arr[i][j];  }  }  p[j] = pow(temp, (double)1 / 3);  }  ofstream out(fileName, ios::app);  out << endl << "Array 2:" << endl;  for (int i = 0; i < n; i++) {  out << p[i] << ' ';  }  delete p;  }  void sqr(char \*fileName, double \*\*arr, int m, int n) {  double \*\*arr1 = new double\*[m];  for (int i = 0; i < m; i++) {  arr1[i] = new double[n];  for (int j = 0; j < n; j++) {  if ((i + j) % 2 == 0) {  arr1[i][j] = arr[i][j] \* arr[i][j];  }  else {  arr1[i][j] = arr[i][j];  }  }  }  ofstream out(fileName, ios::app);  out << endl << "Array 1:" << endl;  for (int i = 0; i < m; i++) {  for (int j = 0; j < n; j++) {  out << arr1[i][j] << ' ';  }  out << endl;  }  for (int i = 0; i < m; i++)  {  delete[] arr1[i];  }  delete[] arr1;  }  void main()  {  int i = 0;  int j = 0;  int count = 0;  int m = 0;  const int n = 3;  double\*\* arr=readFromFile("data.txt", count, n, m);  initial("results.txt", arr, m, n);  sqr("results.txt", arr, m, n);  root("results.txt", arr, m, n);  for (int i = 0; i < m; i++)  {  delete[] arr[i];  }  delete[] arr;  system("pause");  } |

Файл data.txt у каталозі програми:

|  |
| --- |
| 2 34 3  4 5 6  7 8 9  6 5 2  7 4 |

Приклад виконання програми (файл results.txt):

|  |
| --- |
| Initial Array:  2 34 3  4 5 6  7 8 9  6 5 2  7 4 0  Array 1:  4 34 9  4 25 6  49 8 81  6 25 2  49 4 0  Array 2:  1.25992 1.5874 0 |

Висновок

Структури це складені типи. Користувач сам задає, які данні входять до структури. Складені списки є більш гнучкими у використанні ніж масиви, хоча й не простими про освоєнні. Їхня перевага у тому, що вони дозволяють додавати елементи будь де у межах списку.