პოინტერები

**დავალება 1:**

დაწერეთ C ++ პროგრამა რომელიც მოახდენს 3 ელემენტიანი მთელი რიცხვების მასივის გამოტანას მიმთითებლების გამოყენებით (თითოეული მასივის ელემენტიც გამოიტანეთ ასევე)

**დავალება 2:**

დაწერეთ C ++ პროგრამა რომელიც მოახდენს ორი ცვლადისათვის ადგილის გაცვლას , შექმენით 3 მეთოდი რომ მოახდინოთ დემონსტრირება : pass by value, pass by reference, pass by pointer - ის

დინამიური მასივი

**დავალება 3:**

დაწერეთ პროგრამა, რომელიც:

\* სთხოვს მომხმარებელს რამდენი სახელის შეტანა სურს.

\* დინამიურად გამოყოფს std :: სტრინგ მასივს.

\* სთხოვს მომხმარებელს თითოეული სახელს მასივში ჩასაწერად.

\* იყენებს std :: sort მეთოდს სორტირებისათვის

\* ბეჭდავს სახელების დალაგებულ სიას.

std :: string აქვს შედარების ოპერატორების მხარდაჭერა შესაბამისად შეგიძლიათ გამოიყენოთ < და >, შესაბამისად არ გჭირდებათ სტრინგების ხელით შედარება ხელით.

თქვენი აუთფუთი უნდა შეესაბამებოდეს ამას:

How many names would you like to enter? 5

Enter name #1: Jason

Enter name #2: Mark

Enter name #3: Alex

Enter name #4: Chris

Enter name #5: John

Here is your sorted list:

Name #1: Alex

Name #2: Chris

Name #3: Jason

Name #4: John

Name #5: Mark

პასუხები

Exercise 1:

// Exercises: Pointers

// Exercise 1

#include <iostream>

using namespace std;

const int MAX = 3;

int main () {

int var[MAX] = {10, 100, 200};

int \*ptr;

// let us have address of array (in fact first element) in pointer.

ptr = var;

for (int i = 0; i < MAX; i++) {

cout << "Address of var[" << i << "] = ";

cout << ptr << endl;

cout << "Value of var[" << i << "] = ";

cout << \*ptr << endl;

// point to next location

ptr++;

}

return 0;

}

Exercise 2:

//Filename: swap.cpp

#include <string.h>

#include <iostream>

**using** **namespace** std;

**void** **swap1** (**int** x, **int** y);

**void** **swap2** (**int**& x, **int**& y);

**void** **swap3** (**int\*** x, **int\*** y);

**int** **main** ()

{

**int** a = **5**;

**int** b = **9**;

cout << "This program can not swap 2 values using value parameters." << endl;

cout << "Values before the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

swap1(a, b); // code that calls the function

cout << "Values after the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

cout << "This program swaps 2 values using reference parameters." << endl;

cout << "Values before the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

swap2(a, b); // code that calls the function

cout << "Values after the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

cout << "This program swaps 2 values using pointer parameters." << endl;

cout << "Values before the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

swap3(&a,&b); // code that calls the function

cout << "Values after the swap:" << endl;

cout << "a= " << a << " b= " << b << endl;

}

// passing by value

**void** **swap1** (**int** x, **int** y)

{

**int** temp;

temp = x;

x = y;

y = temp;

**return**;

} // end swap ცვლილება არ მოხდება ოპერაცია კოპიოებზე

// passing by reference

**void** **swap1** (**int**& x, **int**& y)

{

**int** temp;

temp = x;

x = y;

y = temp;

**return**;

} // end swap ცვლილება მოხდება ოპერაცია ორიგინალებზე

// passing by pointer

**void** **swap1** (**int**\* x, **int**\* y)

{

**int** temp;

temp = \*x;

\*x = \*y;

\*y = temp;

**return**;

} // end swap ცვლილება მოხდება ოპერაცია ორიგინალებზე

Exercise 4:

#include <algorithm> // For sort

#include <iostream>

#include <string>

using namespace std;

int getNameCount()

{

cout << "How many names would you like to enter? ";

int length{};

cin >> length;

return length;

}

// სთხოვს მომხმარებელს შემოიტანოს სახელები და ინახავს მასივში

void getNames(string\* names, int length)

{

for (int i = 0; i < length; ++i)

{

cout << "Enter name #" << i + 1 << ": ";

cin >> names[i];

}

}

// ბეჭდავს სორტირებულ სახელებს

void printNames(string\* names, int length)

{

cout << "\nHere is your sorted list:\n";

for (int i{ 0 }; i < length; ++i)

cout << "Name #" << i + 1 << ": " << names[i] << '\n';

}

int main()

{

int length = getNameCount();

// ვქმნით მასივს რომელშიც ვინახავთ სახელებს

auto \* names = new string[length];

getNames(names, length);

// ვასორტირებთ

sort(names, names + length);

printNames(names, length);

// ვშლით მასივს

delete[] names;

return 0;

}