МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СХІДНОУКРАЇНСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

імені ВОЛОДИМИРА ДАЛЯ

Звіт

з лабораторної роботи № 3

# з теми: «использование перегруженных функций и работа с указателями на функции в языке С++»

Виконав студент:

Група КН-18д

Скороход С.Г.

Дата 09.05.2020

Варіант №3

**м. Северодонецьк**

2020

## Цель работы

Научиться использовать указатели на функции и перегруженные функции.

Створюемо глобальну константну змінну для розміру масивів

const int Size = 10;

Шаблонна функція для заповнення масивів

template<class T>

void createArr(T \*arr)

{

cout << "Enter mas" << endl;

T \*p = new T;

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

{

cin >> \*p;

arr[i] = \*p;

}

}

Шаблонна функція для виводу масивів

template<class T>

void writeArr(T \*arr)

{

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

{

cout << arr[i]<<tab;

}

cout << endl;

}

Перегружені функції заміни елементів

void swap(string& left, string& right)

{

string buffer = left;

left = right;

right = buffer;

}

void swap(int& left, int& right)

{

int buffer = left;

left = right;

right = buffer;

}

Перегружені функції які визначають чи більший перший елемент за другий

bool compare\_greater(string left, string right)

{

if (left > right)

return true;

else

return false;

}

bool compare\_greater(int left, int right)

{

if (left > right)

return true;

else

return false;

}

Перегружені функції сортування масивів бульбашковим методом

void sort(string \*arr, bool(\*greater)(string left, string right), void(\*swp)(string& left, string& right))

{

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

{

for (int j = Size - 1; j > i; j--)

{

if (greater(arr[j - 1],arr[j]))

{

swp(arr[j - 1], arr[j]);

}

}

}

}

void sort(int \*arr, bool(\*greater)(int left, int right), void(\*swp)(int& left, int& right))

{

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

{

for (int j = Size - 1; j > i; j--)

{

if (greater(arr[j - 1], arr[j]))

{

swp(arr[j - 1], arr[j]);

}

}

}

}

Перегружені функції сортування масивів методом прямого включення

void sortP(string \*arr, bool(\*greater)(string left, string right))

{

for (int i = 1; i < Size; i++)

{

int index = i;

string value = arr[i];

string low1 = value;

transform(low1.begin(), low1.end(), low1.begin(), tolower);

string low2 = arr[index - 1];

transform(low2.begin(), low2.end(), low2.begin(), tolower);

while ((index > 0) && (greater(low2,low1)))

{

arr[index] = arr[index - 1];

index--;

if(index != 0) transform(arr[index - 1].begin(), arr[index - 1].end(), low2.begin(), tolower);

}

arr[index] = value;

}

}

void sortI(int \*arr, bool(\*greater)(int left, int right))

{

for (int i = 1; i < Size; i++)

{

int value = arr[i];

int index = i;

while ((index > 0) && (greater(arr[index - 1],value)))

{

arr[index] = arr[index - 1];

index--;

}

arr[index] = value;

}

}

Загальний вид коду

#include <iostream>

#include <string>

#include <algorithm>

#include <cctype>

#define tab "\t"

//#define TIPE //не в комите работа с целыми числами, иначе со стрингом

#define LR3 // Вариант 3 - не в комите работа с целыми числами, иначе со стрингом

using namespace std;

const int Size = 10;

template<class T>

void createArr(T \*arr);

template<class T>

void writeArr(T \*arr);

void swap(string& left, string& right);

void swap(int& left, int& right);

bool compare\_greater(string left, string right);

bool compare\_greater(int left, int right);

void sort(string \*arr, bool(\*greater)(string left, string right), void(\*swp)(string& left, string& right));

void sort(int \*arr, bool(\*greater)(int left, int right), void(\*swp)(int& left, int& right));

void sortP(string \*arr, bool(\*greater)(string left, string right));

void sortI(int \*arr, bool(\*greater)(int left, int right));

int main()

{

#ifdef TIPE

cout << "INTEGER" << endl;

int \*mas = new int[Size];

createArr(mas);

writeArr(mas);

sort(mas, compare\_greater, swap);

writeArr(mas);

delete[]mas;

cout << "STRING" << endl;

string \*str = new string[Size];

createArr(str);

writeArr(str);

sort(str, compare\_greater, swap);

writeArr(str);

delete[]str;

#endif // TIPE

#ifdef LR3

cout << "INTEGER" << endl;

int \*mas = new int[Size];

createArr(mas);

writeArr(mas);

sortI(mas,compare\_greater);

writeArr(mas);

delete[]mas;

cout << "STRING" << endl;

string \*str = new string[Size];

createArr(str);

writeArr(str);

sortP(str, compare\_greater);

writeArr(str);

delete[]str;

#endif //

}

//////Заполнение массивов

template<class T>

void createArr(T \*arr)

{

cout << "Enter mas" << endl;

T \*p = new T;

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

{

cin >> \*p;

arr[i] = \*p;

}

}

/////// Вывод массивов

template<class T>

void writeArr(T \*arr)

{

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

{

cout << arr[i]<<tab;

}

cout << endl;

}

//// Свапы елементов

void swap(string& left, string& right)

{

string buffer = left;

left = right;

right = buffer;

}

void swap(int& left, int& right)

{

int buffer = left;

left = right;

right = buffer;

}

///// Определение больше ли первый елемент

bool compare\_greater(string left, string right)

{

if (left > right)

return true;

else

return false;

}

bool compare\_greater(int left, int right)

{

if (left > right)

return true;

else

return false;

}

/////// Сортировка по возрастанию

void sort(string \*arr, bool(\*greater)(string left, string right), void(\*swp)(string& left, string& right))

{

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

{

for (int j = Size - 1; j > i; j--)

{

if (greater(arr[j - 1],arr[j]))

{

swp(arr[j - 1], arr[j]);

}

}

}

}

void sort(int \*arr, bool(\*greater)(int left, int right), void(\*swp)(int& left, int& right))

{

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

{

for (int j = Size - 1; j > i; j--)

{

if (greater(arr[j - 1], arr[j]))

{

swp(arr[j - 1], arr[j]);

}

}

}

}

//// сортировка прямым включением

void sortP(string \*arr, bool(\*greater)(string left, string right))

{

for (int i = 1; i < Size; i++)

{

int index = i;

string value = arr[i];

string low1 = value;

transform(low1.begin(), low1.end(), low1.begin(), tolower);

string low2 = arr[index - 1];

transform(low2.begin(), low2.end(), low2.begin(), tolower);

while ((index > 0) && (greater(low2,low1)))

{

arr[index] = arr[index - 1];

index--;

if(index != 0) transform(arr[index - 1].begin(), arr[index - 1].end(), low2.begin(), tolower);

}

arr[index] = value;

}

}

void sortI(int \*arr, bool(\*greater)(int left, int right))

{

for (int i = 1; i < Size; i++)

{

int value = arr[i];

int index = i;

while ((index > 0) && (greater(arr[index - 1],value)))

{

arr[index] = arr[index - 1];

index--;

}

arr[index] = value;

}

}

Висновок: було створено програму для сортування масивів цілих чисел та масивів символьних строк бульбашковим та методом прямого включення.