## Лабораторная работа №6

## СОРТИРОВКА В МАССИВАХ

**Цели работы**

Приобретение навыков составления и отладки программ:

* Изучение алгоритмов сортировки в массивах,
* Разработка многомодульных программ,
* Построение динамических массивов,
* Отработка принципов передачи массивов в качестве аргументов функций.

**Постановка задачи**

* Составить программу сортировки массива методом пирамидальной сортировки.
* В программе должен использоваться динамический массив.
* При инициализации исходного массива предусмотреть режим «ручного» ввода с клавиатуры (при размере массива n<=10) и режим генерации псевдослучайных чисел (при размере массива n>10).
* Инициализацию исходного массива, сортировку массива, распечатку массива оформить в виде отдельных функций.

**Описание алгоритма**

Рис. 1. Схема алгоритма программы

Начало

Инициализация mas, sizemas;

Ввод sizemas;

2/2

2/1

1/2

1/1

cout << "Массив с примененной пирамидальной сортировкой";

false

randomas(mas);

true

sizemas>=10

true

sizemas<1

false

cout<<”Введите корректное sizemas”;

handmas(mas);

build\_maxheap(mas,sizemas);

heapsort(mas, sizemas);

output(mas);

Конец

Рис. 2. Схема алгоритма программы

Начало

Инициализация \*mas, sizemas, randomansw;

i = 1;

false

i <=sizemas;

i++;

mas[i] = rand() % 1000;

Конец

true

Рис. 3. Схема алгоритма функции randomas()

Начало

Инициализация \*mas, sizemas;

cout << "Элемент массива [" << i << "]= ";

i++;

i = 1;

Конец

i <=sizemas;

false

true

cin>>mas[i];

Рис. 4. Схема алгоритма функции handomas()

Начало

Инициализация \*mas, sizemas;

Конец

i++;

i <=sizemas;

i = 1;

false

true

cout << "Элемент массива [" << i << "]= " << mas[i] << endl;

Рис. 5. Схема алгоритма функции output()

Начало

Инициализация \*mas, sizemas;

i = sizemas/2;

false

i>=1

Конец

i--;

max\_heapify(mas, i, sizemas);

true

Рис. 6. Схема алгоритма функции build\_maxheap()

Конец

i--;

temp = mas[i]; mas[i] = mas[1]; mas[1] = temp; max\_heapify(mas, 1, i - 1);

Начало

Инициализация \*mas, sizemas, temp;

i = sizemas;

i>=2

false

true

Рис. 7. Схема алгоритма функции heapsort()

**Описание программы**

Программа инициализирует массив размера sizemas (sizemas – глобальная переменная) и указатель на него. Далее пользователю предлагается ввести размер массива. Он проходит проверку на значение и тип данных посредством функции cin.fail. В случае, если размер массив превышает 10 – массив заполняется датчиком псевдослучайных чисел rand(), иначе – заполняется с клавиатуры. Оба метода реализованы подпрограммами – randomas() и handmas().

Далее объявленный массив сортируется пирамидальным методом, реализованным тремя подпрограммами: build\_maxheap() – создание «кучи» из массива, heapsort() – сортировка «кучи» и внутренней max\_heapify().

Отсортированный массив выводится на экран при помощи подпрограммы output().

**Текст исходного кода программы с комментариями**

#include <iostream>

#include <cstdlib>

using namespace std;

int sizemas; //глобальный размер массива

void randomas(int \*mas) { //заполение массива датчиком псевдослучайных чисел

int randomansw;

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

{

mas[i] = rand() % 1000;

}

cout << "Посмотреть массив - 0. Пропустить - 1" << endl;

cin >> randomansw;

if (randomansw > 1) cout << "Неверно" << endl;

while (randomansw > 1) {

cout << "Выбор: ";

cin >> randomansw;

if (randomansw > 1) cout << "Неверно" << endl;

}

switch (randomansw) {

case 0:

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

{

cout << "Элемент массива [" << i << "]= " << mas[i] << endl;

}

break;

case 1:

break;

}

}

void handmas(int \*mas) { //ручной ввод массива

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

{

cout << "Элемент массива [" << i << "]= ";

cin >> mas[i];

if (mas[i] > 1000) cout << "Неверный размер: < 1.000." << endl;

while (mas[i] > 1000)

{

cout << "Элемент массива [" << i << "]= ";

cin >> mas[i];

if (mas[i] > 1000) cout << "Неверный размер: < 1.000." << endl;

}

if (mas[i] <1) cout << "Неверный размер; > 0." << endl;

while (mas[i] < 1)

{

cout << "Элемент массива [" << i << "]= ";

cin >> mas[i];

if (mas[i] > 1000000) cout << "Неверный размер: > 0." << endl;

}

}

}

void output(int \*mas){ //вывод полученного массива

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

{

cout << "Элемент массива [" << i << "]= " << mas[i] << endl;

}

}

void max\_heapify(int \*mas, int i, int sizemas) //работает с элеиентом и следующим за ним

{

int j, temp;

temp = mas[i];

j = 2\*i;

while (j <= sizemas) //внутри массива

{

if (j < sizemas && mas[j+1] > mas[j])

j = j+1;

if (temp > mas[j])

break;

else if (temp <= mas[j])

{

mas[j/2] = mas[j];

j = 2\*j;

}

}

mas[j/2] = temp;

return;

}

void heapsort(int \*mas, int sizemas) //сортировка

{

int i, temp;

for (i = sizemas; i >= 2; i--)

{

temp = mas[i];

mas[i] = mas[1];

mas[1] = temp;

max\_heapify(mas, 1, i - 1);

}

}

void build\_maxheap(int \*mas, int sizemas) //создаем кучу(пирамиду)

{

int i;

for(i = sizemas/2; i >= 1; i--)

{

max\_heapify(mas, i, sizemas);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int \*mas;

mas = new int[sizemas];

cout << "Введите размер массива: ";

cin >> sizemas;

if (sizemas < 1 || cin.fail())

cout << "Неверно. Введите число от 1." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

while (sizemas<1) {

cout << "Введите размер массива: ";

cin >> sizemas;

if (sizemas<1|| cin.fail()) cout << "Неверно. Введите число от 1." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

}

if (sizemas >= 10)

randomas(mas);

else handmas(mas);

cout << endl;

cout << "Массив с примененной пирамидальной сортировкой" << endl;

cout << endl;

build\_maxheap(mas,sizemas);

heapsort(mas, sizemas);

output(mas);

cout << endl;

system("Pause");

return 0;

}

**Контрольные прогоны программы**

Тест №1 – ввод значения, соответствующего ручному ввода массива. Последующая сортировка полученного массива.

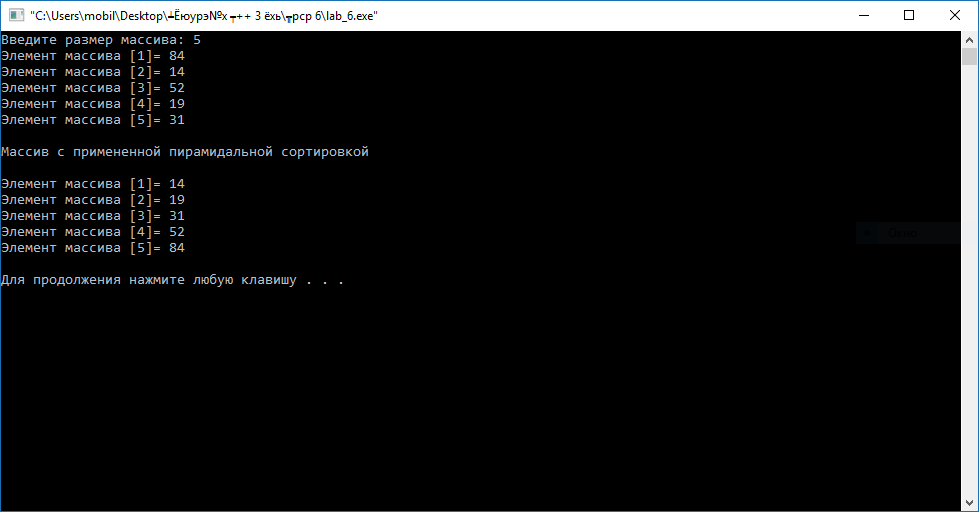


Рис. 8. Результат теста №1

Тест №2 – ввод значения, соответствующего заполнению массива псевдослучайными числами. Последующая сортировка полученного массива.

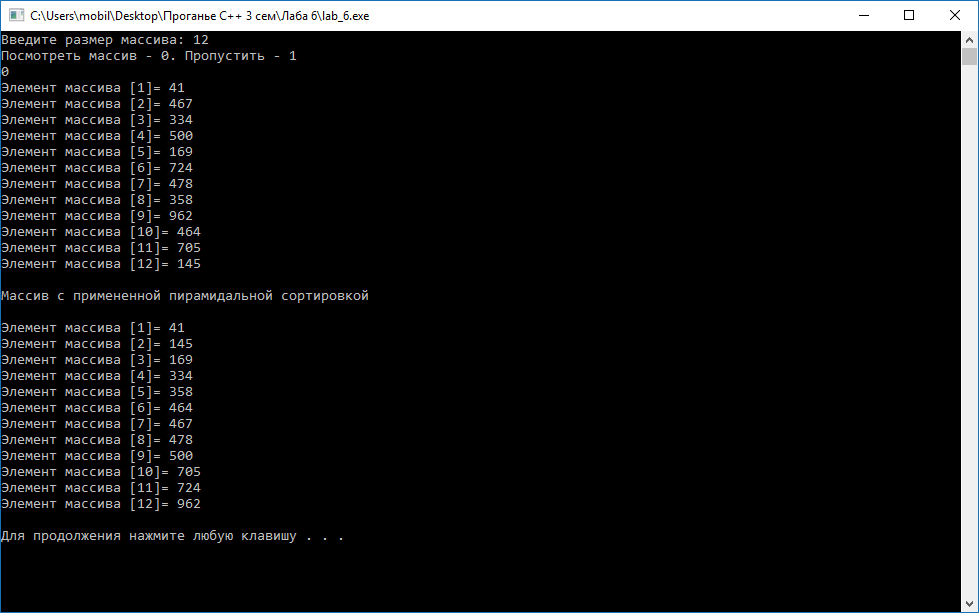


Рис. 9. Результат теста №2

Тест №3 – ввод значения, соответствующего заполнению массива псевдослучайными числами. Последующая сортировка полученного массива.

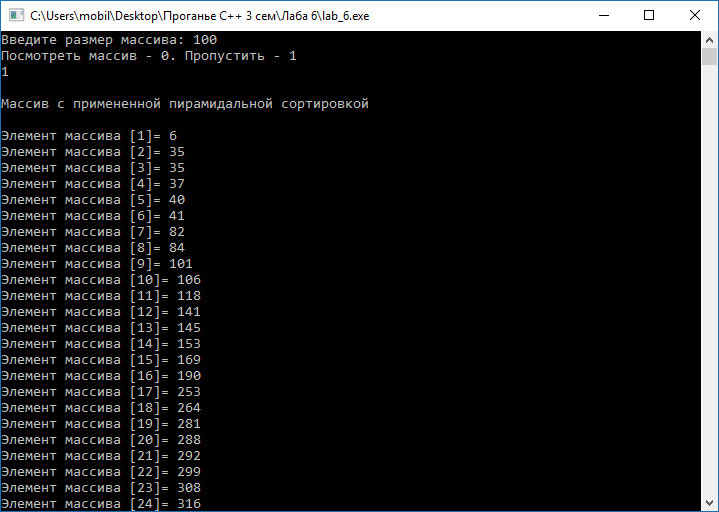


Рис. 10. Результат теста №3

Тест №4 – ввод значения, соответствующего заполнению массива псевдослучайными числами. Последующая сортировка полученного массива.

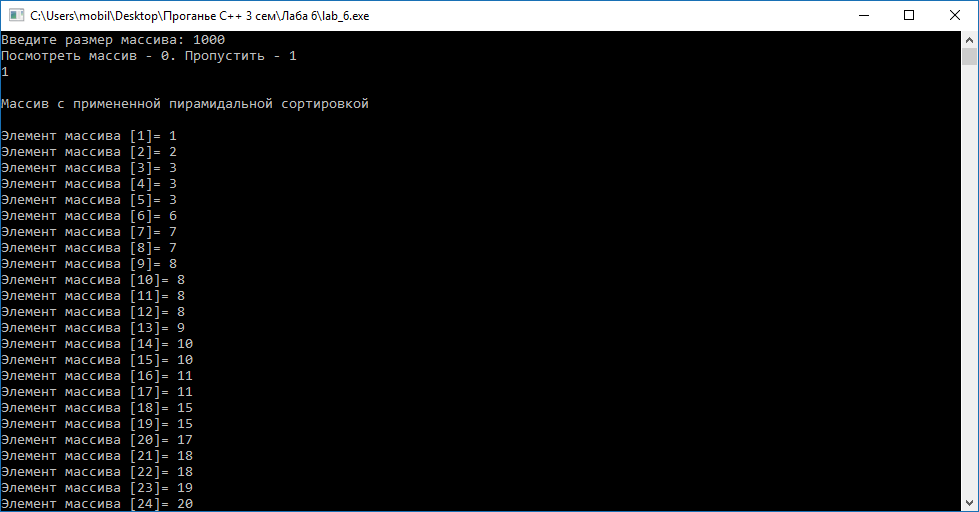


Рис. 11. Результат теста №4

**Выводы**

Данная программа позволила мне освоить работу с динамическими массивами, подпрограммами и сортировкой в С++.

**Список используемых информационных источников**

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня- Санкт-Петербург. Питер, 2006.

2. Интернет-ресурс: http://proprogramming.org/heap-sort-in-c/

3. Интернет-ресурс: http://cybern.ru/heapsortcpp.html

4. Единая система программной документации (ЕСПД) ГОСТ 19.701-90

5. Сыромятников В.П. «Программирование», лекции, МТУ МИРЭА, 2016/2017.