МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2

по дисциплине «Технологии программирования»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2016

1. Задание на лабораторную работу

Используя функции malloc и free реализовать однонаправленный кольцевой список, который содержит внутри себя указатель на массив и 3 символа. Проанализировать дамп памяти и на основе этого сделать вывод о распределении памяти в методе граничных маркеров.

1. Исследование дампа памяти

Первым параметром запрашивается у пользователя размер входного массива. Далее пользователь вводит с клавиатуры номер типа данных массива (1 – целочисленный тип, 2 - вещественный). На основе данных формируется пустой массив заданной размерности и типа. Пользователь вводит желаемое значение К. Пользователь последовательно вводит элементы массива. Следующим этапом является сортировка массива. Массив сортируется при помощи встроенной функции сортировки. Заключительным этапом является вывод на экран произведение К и К+1 элемента отсортированного массива.

1. Исходный код программы

Исходный код программы представлен на рисунке 1.

|  |
| --- |
| #include <iostream>  #include <algorithm>  #include <string>  using namespace std;  bool val(string s, int &n, int a, int b) {  n = atoi(s.c\_str());  if ( (to\_string(n).length() == s.length()) && (n >= a) && (n <= b) ) {  return true;  } else {  return false;  }  }  bool val(string s, float &n, int a, int b) {  n = atof(s.c\_str());  if ( (to\_string(n).length() == s.length()) && (n >= a) && (n <= b) ) {  return true;  } else {  return false;  }  }  template <typename Arr>  void input(Arr \* arr, int count) {  string buf;  cout << "Введите массив: " << endl;  for (int i = 0; i < count; i++) {  cout << "m[" << i << "] = ";  cin >> buf;  val(buf, arr[i], -1000, 1000);  }  }  template <typename Arr>  void print(const Arr \* arr, int count) {  for (int i = 0; i < count; i++) {  cout << arr[i] << " ";  }  cout << endl;  }  int kStat(int\* arr, int count, int k) {  sort(arr, arr + count);  return arr[k] \* arr[k + 1];  }  float kStat(float\* arr, int count, int k) {  sort(arr, arr + count);  return arr[k] \* arr[k + 1];  }  int main(int argc, char const \*argv[]) {  int n;  string buf;  do {  cout << "Введите размер массива: ";  cin >> buf;  } while (!(val(buf, n, 2, 1000)));  int k;  do {  cout << "Введите K(от 1 до N-1): ";  cin >> buf;  } while (!(val(buf, k, 1, n-1)));  k--;  int type;  do {  cout << "Тип массива: \n" << "1) Целочисленный (integer)\n" << "2) Вещественный (float)" << endl;  cin >> buf;  } while (!(val(buf, type, 1, 2)));  switch (type) {  case 1: {  int \*arr = new int [n];  input(arr, n);  print(arr, n);  cout << kStat(arr, n, k) << endl;  break;  }  case 2: {  float \*arr = new float [n];  input(arr, n);  print(arr, n);  cout << kStat(arr, n, k) << endl;  break;  }  default : {  cout << "Неверный тип. Попробуйте еще раз." << endl;  }  }  return 0;  } |

Рисунок 1 – Исходный код программы

1. Вывод

В ходе выполнения лабораторной работы были полученные необходимые знания базового синтаксиса языка программирования С++. Были изучены шаблоны функции, перегрузки функции, декораторы. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения различных технологий программирования и языка С++.