ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

ЛАБОРАТОРНАЯ РАБОТА № 5

Основы алгоритмизации и программирования

РАЗРАБОТКА ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ

ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

Выполнил студент группы 782:

Ванин Никита Александрович

Томск 2020

**Цель работы:**

Освоить правила объявления функций, ввод и вывод одномерных массивов, правила работы с указателями, методику работы с пользовательскими функциями.

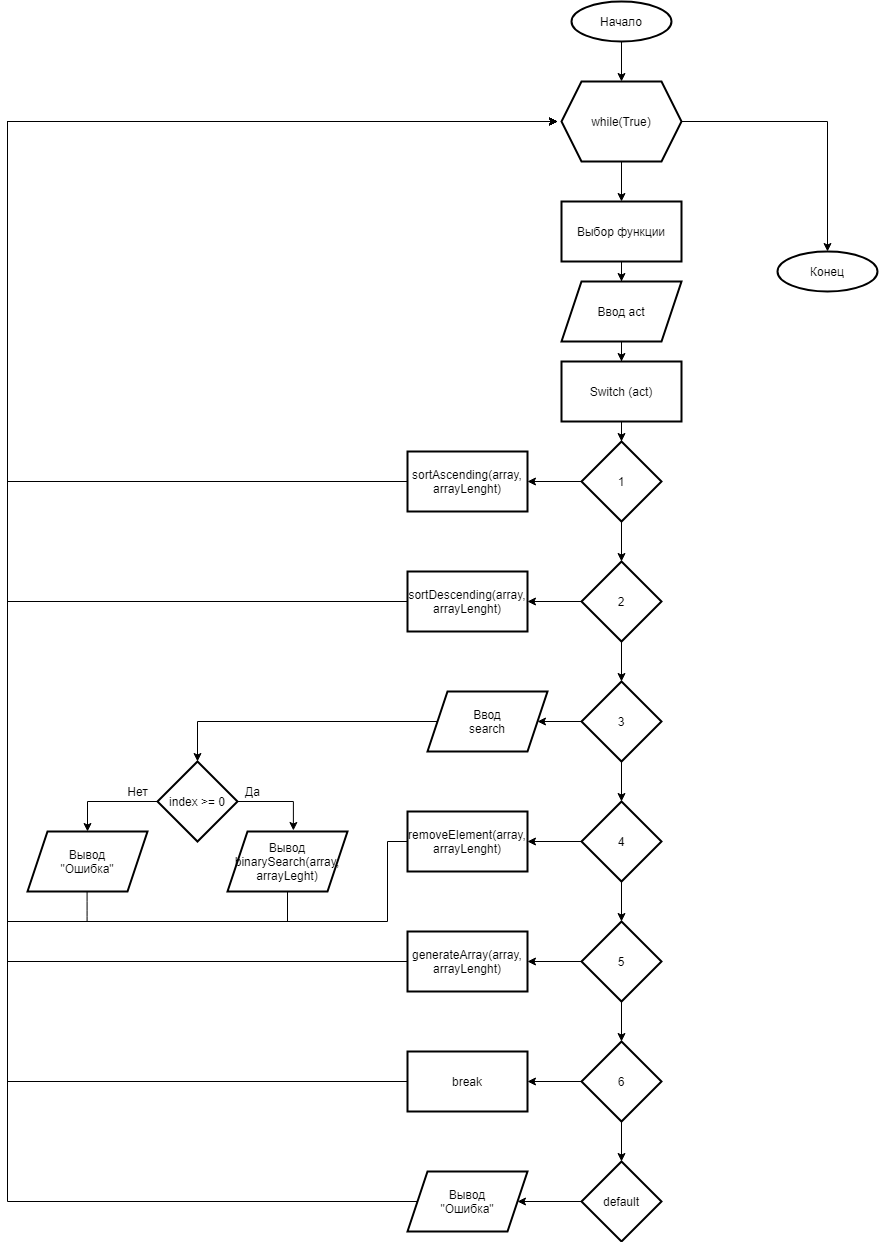
**Содержание работы:**

1. Разработать алгоритм по предложенному варианту.
2. Описать алгоритм с помощью блок-схем. При выполнении преобразований не использовать дополнительные массивы.
3. Составить программу, отладить ее и выполнить.
4. Оформить отчет по проделанной работе, в том числе представить:
   * Схемы алгоритма, оформленную в соответствии с требованиями ГОСТа.
   * Листинг программы на языке С. Текст программы должен содержать комментарии (условие задачи, фамилию, номер группы и другую дополнительную информацию).
   * Результаты выполнения программы.

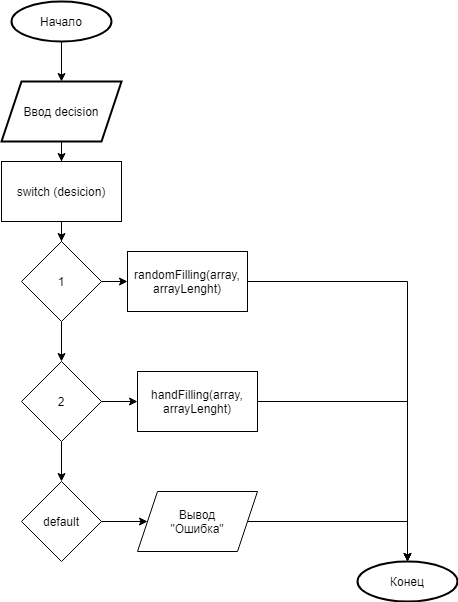
**Объем работы**: 8 часов.

1. Блок-схемы

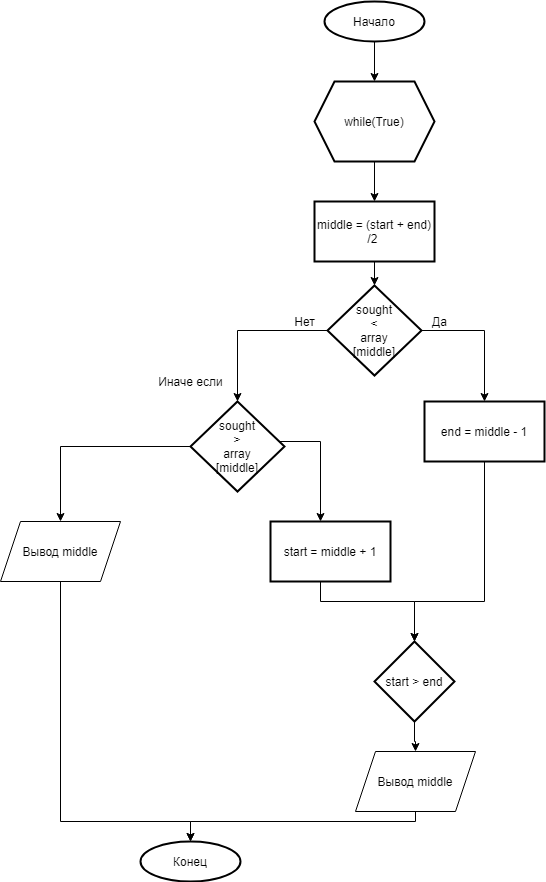
1.1Функция main()



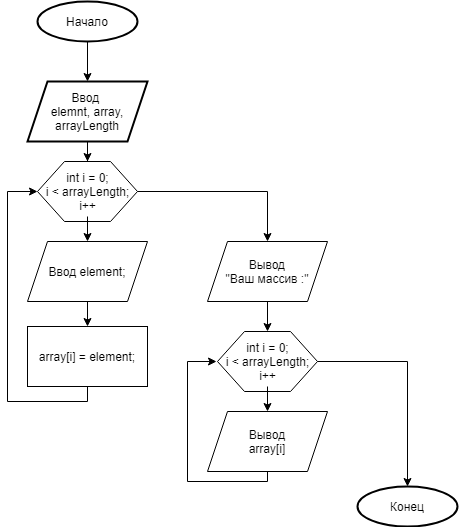
1.2 Функция generateArray()



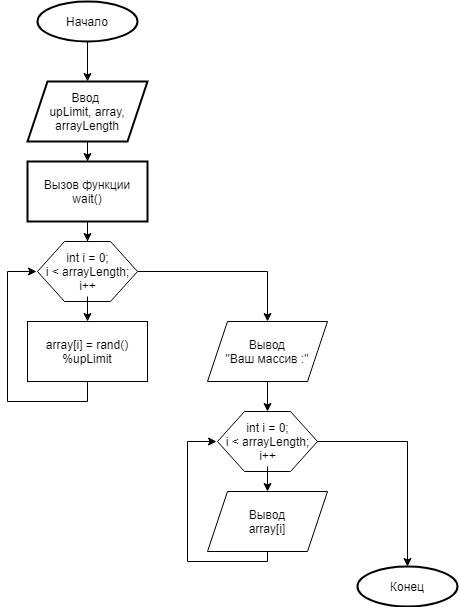
1.3 Функция binarySearch()



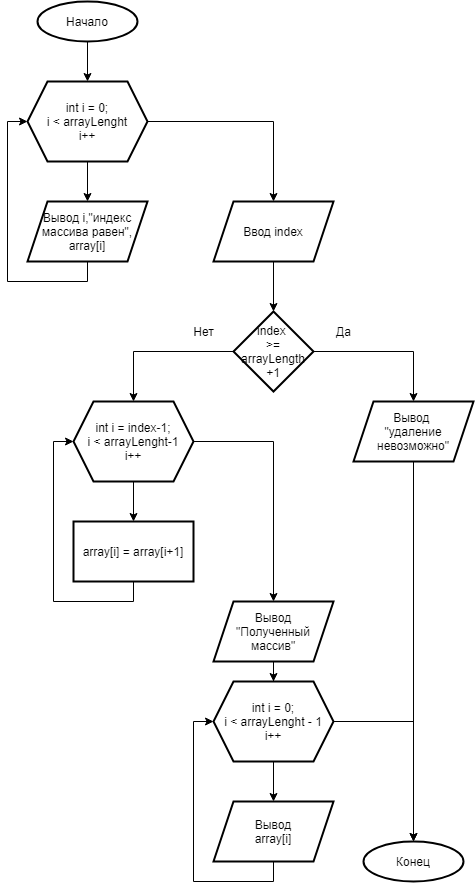
1.4 Функция handFilling()



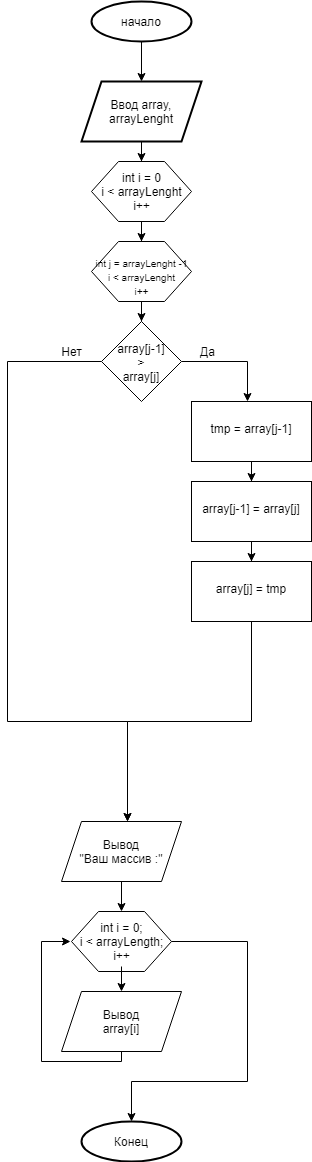
1.5 Функция randomFilling()



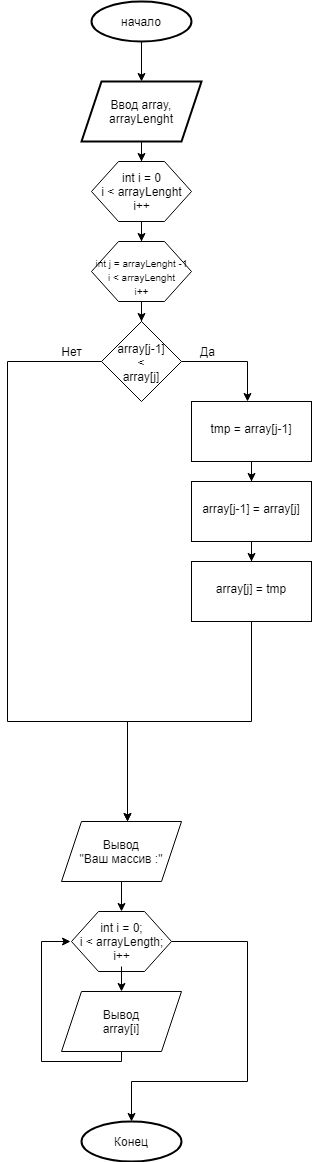
1.6 Функция removeElement()



1.7 Функция sortAscending()



1.8 Функция sortDescending()



2. Листинг кода

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int randomFilling(int array[], int arrayLenght) { //заполнение массива случайными числами

int upLimit;

printf("Введите верхний предел генерации чисел: ");

scanf("%i", &upLimit);

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

array[i] = rand() % upLimit;

}

printf("Ваш массив: ");

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

printf("%i, ", array[i]);

}

printf("\n");

return 0;

}

int handFilling(int array[], int arrayLenght) { //заполнение массива вручную

int element;

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

printf("Введите %i элемент массива: ", i+1);

scanf("%i", &element);

array[i] = element;

}

printf("Ваш массив: ");

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

printf("%i, ", array[i]);

}

printf("\n");

}

int sortAscending (int array[], int arrayLenght){ //Сортировка пузырьком по возрастанию

for (int i = 0; i < arrayLenght - 1; i++) {

for (int j = (arrayLenght - 1); j > i; j--) { //крутим массив

if (array[j - 1] > array[j]) { // если элемент больше

int tmp = array[j - 1]; // обмен значениями

array[j - 1] = array[j];

array[j] = tmp;

};

}

}

printf("Ваш массив: ");

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

printf("%i, ", array[i]);

}

printf("\n");

}

int sortDescending(int array[], int arrayLenght) { //Сортировка пузырьком по убыванию

for (int i = 0; i < arrayLenght - 1; i++) {

for (int j = (arrayLenght - 1); j > i; j--) {

if (array[j - 1] < array[j]) { //меняем знак, чтобы все пошло в другую сторону

int tmp = array[j - 1];

array[j - 1] = array[j];

array[j] = tmp;

};

}

}

printf("Ваш массив: ");

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

printf("%i, ", array[i]);

}

printf("\n");

}

int binarySearch(int array[], int end, int sought) { //бинарный поиск

int middle;

int start = 0;

while (1) {

middle = (start + end) / 2;

if (sought < array[middle]) { //если меньще, чем середина

end = middle - 1; // меняем границы поиска

}

else if (sought > array[middle]) { //если больше, чем середина

start = middle + 1; // меняем границы поиска

}

else { // если не больше и не меньше, то возврат middle

return middle; // вовзрат угаданного числа

}

if (start > end) { // если чисел между границами не остальность

return -1;

}

}

}

int removeElement(int array[], int arrayLenght) { //удаление элемента

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

printf("%i индекс массива имеет значение %i\n", i+1, array[i]); //просмотр всех элементов в массиве

}

printf("Выберите индекс значения которого вы хотите удалить: ");

int index;

scanf("%i", &index);

if (index >= arrayLenght + 1) { //если введенный индекс больше размера массива

printf("Удаление невозможно. Вы ввели несуществющий индекс.\n");

}

else {

for (int i = index-1; i < arrayLenght-1; i++) { //перебор массива, до значения index-1

array[i] = array[i+1]; //обмен значениями элементов // при таком обмене значение array[index] уйдет за пределы массива

}

}

printf("Полученный массив: ");

for (int i = 0; i < arrayLenght-1; i++) {

printf("%i ", array[i]);

}

printf("\n");

}

int generateArray(int array[], int arrayLenght) {

printf("Как вы хотите заполнить массив?\n1 - рандомно,\n2 - вручную,\n0 - выход из программы.\nВаш выбор: ");

int decision, size;

scanf("%i", &decision);

switch (decision) {

case 0:

exit(0);

break;

case 1:

randomFilling(array, arrayLenght);

break;

case 2:

handFilling(array, arrayLenght);

break;

default:

printf("Ну что же вы, следуйте указаниям программы. Попробуйте сначала.\n");

break;

}

return 0;

}

int main() {

system("chcp 65001 && cls");

srand(time(NULL));

printf("добро пожаловать в пятую лабу!\n");

int arrayLenght;

printf("Введите размер массива: ");

scanf("%i", &arrayLenght);

int array[arrayLenght];

generateArray(array, arrayLenght);

int act;

while (1) {

printf("-------------------------\n");

printf("Выберите действие:\n");

printf("-------------------------\n");

printf("1. Сортировка массива по возрастанию\n"); // DA

printf("2. Сортировка массива по убыванию\n"); // DA

printf("3. Бинарный поиск элемента в массиве\n"); // NET

printf("4. Удаление элемента из массива\n"); // DA

printf("5. Повтор генерации массива\n"); // DA

printf("0. Выход из программы.\n");

printf("Ваш выбор: ");

scanf("%i", &act);

switch (act) {

case 0:

exit(0);

break;

case 1:

sortAscending(array, arrayLenght);

break;

case 2:

sortDescending(array, arrayLenght);

break;

case 3:

sortAscending(array, arrayLenght);

printf("Введите число, которое вы хотите найти: ");

int search;

scanf("%i", &search);

int index;

index = binarySearch(array, arrayLenght, search); // binarySearch возвращает значение в index

if (index >= 0)

printf("Указанное число находится в ячейке с индексом: %i\n", index);

else

printf("В массиве нет такого числа!\n");

break;

case 4:

removeElement(array, arrayLenght);

break;

case 5:

generateArray(array, arrayLenght);

break;

default:

printf("Ну что же вы, следуйте указаниям программы. Попробуйте сначала.");

}

}

}

3. Пример выполнения программы

