Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №7.1**

Дисциплина: основы алгоритмизации и программирования

Тема: перегрузка функций

Вариант 23

Выполнил:

Студент группы ИВТ-20-2б

Брейкин Алексей Дмитриевич

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2020**

**Постановка задачи**

Написать перегруженные функции и программу, которая их вызывает.

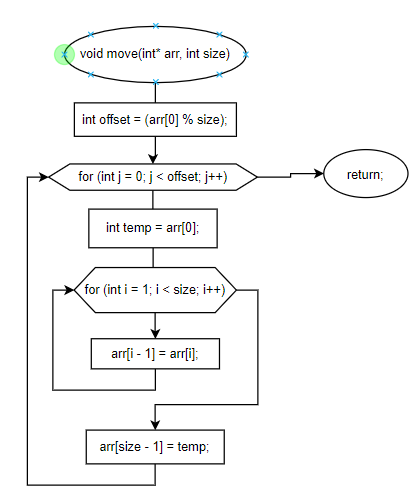
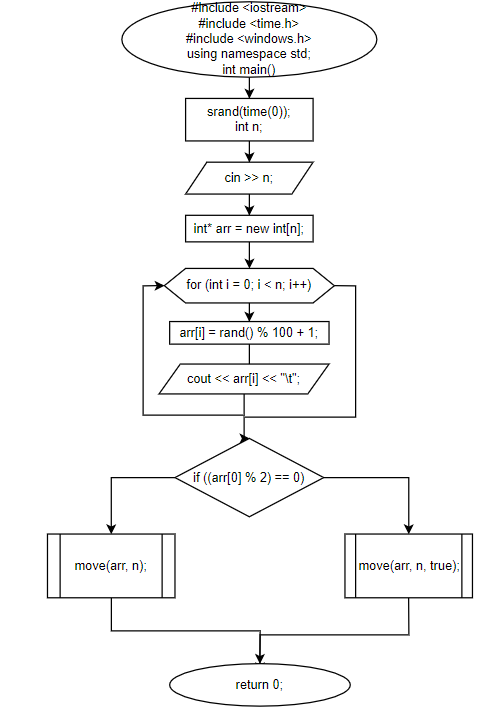
а) для массива, начинающегося на четное число выполняет циклический сдвиг влево на количество элементов равное первому элементу массива.

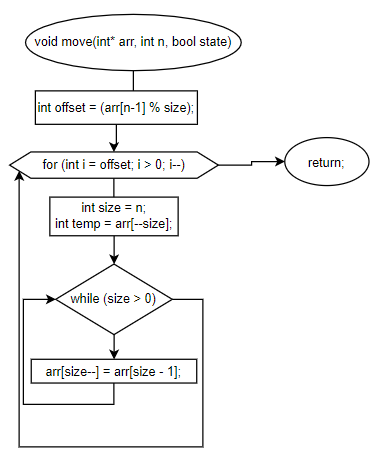
б) для массива, начинающегося на нечетное число выполняет циклический сдвиг вправо на количество элементов равное последнему элементу массива.

**Анализ задачи**

1. Запрашиваем у пользователя нужное количество элементов в массиве;
2. Создаём динамический массив на это число элементов и с помощью цикла и генератора случайных чисел заполняем его;
3. Выводим элементы полученного массива на экран с помощью цикла;
4. Если первый элемент массива чётный, то вызываем функцию move, а в качестве аргументов передаём массив и количество элементов в нём. Эта функция будет выполнять циклический сдвиг влево на то количество позиций, которому равен первый элемент массива. Циклический сдвиг будет выполняться с помощью двух циклов. Внешний цикл будет от 1 до первого элемента массива. А во вложенном цикле будет выполняться сам сдвиг такой, что a[j-1] = a[j]. Важно перед началом вложенного цикла сохранить первый элемент массива, чтобы потом поменять его местами с последним.
5. Если первый элемент массива нечётный, то вызываем функцию move с теми же аргументами, что и в п.4, но в этот раз в конец добавляем флаг true. Это даст понять, что мы вызываем перегрузку функции move с 3 аргументами. В ней будет выполняться циклический сдвиг вправо на количество элементов, равное последнему элементу массива. Алгоритм сдвига тот же, что и в п. 4.
6. Выводим получившийся массив

**Блок-схема**



****

**Исходный код**

#include <iostream>

#include <time.h>

#include <windows.h>

using namespace std;

void move(int\* arr, int size) {

int offset = (arr[0] % size);

for (int j = 0; j < offset; j++) {

int temp = arr[0];

for (int i = 1; i < size; i++) {

arr[i - 1] = arr[i];

}

arr[size - 1] = temp;

}

}

void move(int\* arr, int n, bool state) {

int offset = (arr[n - 1] % n);

for (int i = offset; i > 0; i--) {

int size = n;

int temp = arr[--size];

while (size > 0) arr[size--] = arr[size - 1];

arr[0] = temp;

}

}

int GetInt()

{

int res;

if (!(cin >> res) || (cin.peek() != '\n') || res < 1)

{

cin.clear();

while (cin.get() != '\n');

cout << "Введите целое число, которое больше нуля!" << endl;

}

return res;

}

void PrintArray(int\* arr, int n) {

cout << "Индекс: ";

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

cout << i << "\t";

}

cout << endl << "Значение: ";

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

cout << arr[i] << "\t";

}

cout << endl;

}

int main()

{

SetConsoleOutputCP(CP\_UTF8);

srand(time(0));

int n;

for (;;) {

cout << "Введите размер массива: ";

n = GetInt();

if (n > 0) break;

}

int\* arr = new int[n];

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

arr[i] = rand() % 100 + 1;

}

PrintArray(arr, n);

cout << endl;

if ((arr[0] % 2) == 0) {

cout << "Первый элемент чётный -> выполняем циклический сдвиг влево на " << arr[0] << " позиций" << endl;

move(arr, n);

}

else {

cout << "Первый элемент нечётный -> выполняем циклический сдвиг вправо на " << arr[n - 1] << " позиций" << endl;

move(arr, n, true);

}

cout << "Результат" << endl;

PrintArray(arr, n);

return 0;

}

**Скриншоты работы программы**

