



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

“МИРЭА - Российский технологический университет”

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра математического обеспечения и стандартизации
информационных технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2

по дисциплине

«Структуры и алгоритмы обработки данных»

Тема «Абстрактный тип данных и его реализация на одномерном
статическом массиве»

Выполнил студент группы ИКБО-50-23

Русаков М.Ю.

Принял старший преподаватель

Скворцова Л.А.

Москва 2024

Оглавление

1. Условие задачи и задание варианта	3
1.1. Условие задачи	3
1.2. Задание варианта	3
2. АТД задачи	4
3. Разработка программы	6
3.1. Реализация данных АТД	6
3.2. Алгоритмы методов	7
3.3. Таблица тестов операций	9
3.4. Код проекта	10
3.4.1. Файл <code>myArray.cpp</code>	10
3.4.2. Файл <code>main.cpp</code>	13
4. Скриншоты результатов тестирования	14

1. Условие задачи и задание варианта

1.1. Условие задачи

Дано множество из n целых чисел. Дан набор задач (операций), которые требуется выполнить над исходным множеством. Набор задач определен в варианте задания табл. 6.

Разработать и реализовать АТД задачи, по управлению множеством посредством операций, указанных в варианте задания. В АТД включить операции по заполнению исходного множества и отображения множества.

При разработке алгоритмов операций варианта могут быть выявлены дополнительные алгоритмы, например такие: определить является ли число простым, или определить сумму цифр числа, эти алгоритмы надо включить в раздел операций АТД.

1.2. Задание варианта

1. Найти позицию элемента массива значение которого делится на каждую из цифр числа.
2. Вставить в массив новый элемент после элемента, значение которого делится на каждую цифру значения.
3. Удалить из массива все элементы, кратные трем.

2. АТД задачи

АТД myArray

{

Данные (описание свойств структуры данных задачи)

N – максимальное количество элементов в множестве

n – длина массива

arr – список значений элементов массива

Операции (объявления операций)

1. Метод, осуществляющий вывод текущих значений множества

Предусловие: нет

Постусловие: выведенные через пробел элементы массива

Заголовок: `printElements()`

2. Метод, осуществляющий заполнение массива случайными значениями

Предусловие: нет

Постусловие: массив, заполненный случайными значениями

Заголовок: `fillRandomly()`

3. Метод, осуществляющий заполнение массива вручную – с клавиатуры

Предусловие: нет

Постусловие: массив, заполненный значениями, введенными с клавиатуры

Заголовок: `fillManually()`

4. Метод, возвращающий индекс первого элемента, делящегося на каждую из своих цифр. В случае отсутствия такового элемента возвращается -1

Предусловие: нет

Постусловие: число – индекс первого элемента, нацело делящегося на каждую из своих цифр

Заголовок: `getIndex()`

5. Метод осуществляющий вставку элемента newElem на позицию с индексом pos

Предусловие: pos – индекс элемента, на место которого требуется вставить новый элемент, newElem – значение нового элемента

Постусловие: массив arr длиной $n+1$ со вставленным элементом newElem на позиции pos

Заголовок: insert(int pos, int newElem)

6. Метод, осуществляющий вставку нового элемента newElem после элемента с индексом getIndex(). Если getIndex() = -1, вставка производится в начало массива

Предусловие: newElem – значение нового элемента

Постусловие: массив arr длиной $n + 1$ со вставленным элементом newElem на позиции getIndex()+1

Заголовок: getIndexInsert(int newElem)

7. Метод, осуществляющий удаление из массива всех элементов, нацело делящихся на 3

Предусловие: нет

Постусловие: измененный массив arr, содержащий элементы, не делящихся нацело на 3

Заголовок: deleteMultiplesOfThree()

}

3. Разработка программы

3.1. Реализация данных АД

```
#ifndef MYARRAY__H
#define MYARRAY__H

#include <iostream>

struct myDynArray {
    const static int N = 100;
    unsigned int n;
    int arr[N];

    myDynArray(int len);

    void printElements();

    void fillRandomly();

    void fillManually();

    int getIndex();

    void insert(int pos, int newElem);

    void getIndexInsert(int newElem);

    void erase(int pos);

    void deleteMultiplesOfThree();
};

#endif
```

3.2. Алгоритмы методов

Таблица 1 – Псевдокод для метода, реализующего удаление элемента в заданной позиции

void erase(int pos)	
Номер	Инструкция
1	if $\neg(\text{pos} > -1 \wedge \text{pos} < n)$ then
2	Вывод « Невозможно удалить элемент: неверный индекс! »
3	endIf
4	for $i \leftarrow$ to $n - 2$ do
5	arr[i] \leftarrow arr[i + 1]
6	od
7	$n \leftarrow n + 1$

Таблица 2 – Псевдокод для метода, реализующего вставку элемента в заданной позиции

void insert(int pos, int newElem)	
Номер	Инструкция
1	if $\neg(\text{pos} > -1 \wedge \text{pos} < n)$ then
2	Вывод «Невозможно вставить элемент: неверный индекс! »
3	endIf
4	prev \leftarrow arr[pos]
5	arr[pos] \leftarrow newElem
6	for $i \leftarrow$ pos + 1 to $n + 1$ do
7	temp \leftarrow arr[i]
8	arr[i] \leftarrow prev
9	prev \leftarrow temp
10	od
11	$n \leftarrow n + 1$

Таблица 3 – Псевдокод для метода, реализующего поиск первого элемента, делящегося нацело на каждую из своих цифр

void getIndex()	
Номер	Инструкция
1	for i \leftarrow to n – 1 do
2	number \leftarrow arr[i]
3	while number > 0 do
4	if number mod 10 \neq 0 then
5	if arr[i] mod (number mod 10) \neq 0 then
6	break
7	endIf
8	else
9	break
10	endIf
11	number \leftarrow number div 10
12	od
13	if number = 0 then
14	return i
15	endIf
16	return –1

3.3. Таблица тестов операций

Номер	Операция	Входные данные	Результат работы
1	erase	n=7 arr={1, 2, 3, 5, 8, 13, 21} pos=3	n=6 arr={1, 2, 3, 8, 13, 21}
2	erase	n=2 arr={9, 11} pos=2	Ошибка: «Невозможно удалить элемент: неверный индекс!»
3	insert	n=7 arr={1, 2, 3, 5, 8, 13, 21} pos=3 newElem=4	n=8 arr={1, 2, 3, 4, 5, 8, 13, 21}
4	insert	n=5 arr={2, 2, 2, 2, 2} pos=5 newElem=8	Ошибка: «Невозможно вставить в массив: неверный индекс!»
5	getIndex	n=7 arr={13, 17, 31, 53, 15, 13, 21}	4
6	getIndex	n=3 arr={13, 17, 97}	-1

3.4. Код проекта

3.4.1. Файл myArray.cpp

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <random>
#include "myArray.h"

// Конструктор. Инициализация переменной n (длина массива)
myDynArray::myDynArray(int len) {
    n = len;
}

// Вывод текущих элементов массив в консоль
void myDynArray::printElements() {
    for (size_t i{}; i < n; i++) {
        std::cout << arr[i] << ' ';
    }
    std::cout << '\n';
}

// Метод, осуществляющий заполнение массива случайными значениями
void myDynArray::fillRandomly() {
    srand(time(NULL));

    for (size_t i{}; i < n; i++) {
        int num = rand();
        arr[i] = num;
    }
}

// Метод, осуществляющий заполнение массива вручную
void myDynArray::fillManually() {
    for (size_t i{}; i < n; i++) {
        std::cin >> arr[i];
    }
}

/* Метод, возвращающий индекс первого элемента, делящегося на каждую из
своих цифр.
В случае отсутствия такого элемента возвращается -1. */
int myDynArray::getIndex() {
    for (size_t i{}; i < n; i++) {
        int number = arr[i];
```

```

        while (number > 0) {
            if (number % 10 != 0) {
                if (arr[i] % (number % 10) != 0) {
                    break;
                }
            } else {
                break;
            }

            number /= 10;
        }

        if (number == 0) {
            return i;
        }
    }

    return -1;
}

/* Метод осуществляющий вставку элемента newElem на позицию с индексом
pos */
void myDynArray::insert(int pos, int newElem) {
    if (!(pos > -1 && pos < this->n)) {
        std::cout << "Невозможно вставить в массив: неверный индекс!\n";
        return;
    }

    int prev = arr[pos];
    arr[pos] = newElem;

    for (size_t i = pos + 1; i < n + 1; i++) {
        int temp = arr[i];
        arr[i] = prev;
        prev = temp;
    }

    n++;
}

/* Метод, осуществляющий вставку нового элемента newElem после элемента
с индексом getIndex().
Если getIndex() = -1, вставка производится в начало массива */
void myDynArray::getIndexInsert(int newElem) {
    insert(getIndex() + 1, newElem);
}

```

```

/* Метод, осуществляющий удаление элемента с индексом pos */
void myDynArray::erase(int pos) {
    if (!(pos > -1 && pos < this->n)) {
        std::cout << "Невозможно удалить элемент: неверный индекс!\n";
        return;
    }

    for (size_t i = pos; i < this->n - 1; i++) {
        arr[i] = arr[i+1];
    }

    n--;
}

/* Метод, осуществляющий удаление из массива всех элементов, нацело
делящихся на 3*/
void myDynArray::deleteMultiplesOfThree() {
    int buffArr[N]{};
    int index{};

    for (size_t i{}; i < n; i++) {
        if (!(arr[i] % 3 == 0)) {
            buffArr[index++] = arr[i];
        }
    }

    for (size_t i{}; i < index; i++) {
        arr[i] = buffArr[i];
    }

    n = index;

    for (size_t d = index + 1; d < N; d++) {
        arr[d] = 0;
    }
}

```

3.4.2. Файл main.cpp

```
#include "myArray.h"

int main() {
    int n;
    std::cin >> n;
    if (!(n > 0 && n < 100)) {
        return 1;
    }

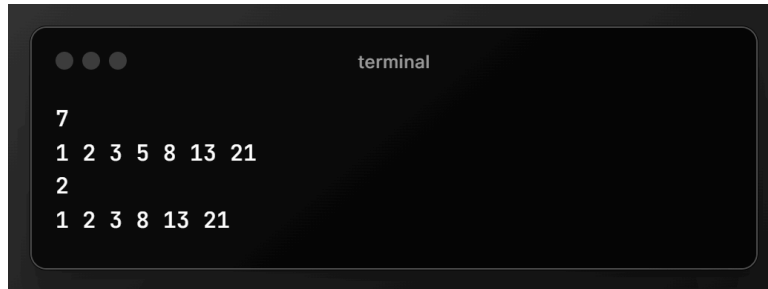
    myDynArray myObj = myDynArray(n);

    myObj.fillManually();

    std::cout << myObj.getIndex();
}
```

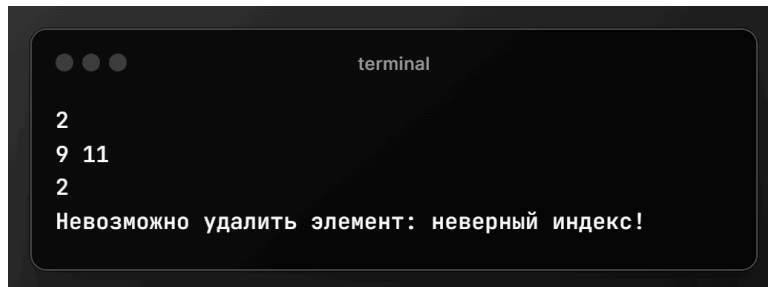
4. Скриншоты результатов тестирования

Комментарий: на скриншотах ниже первое число – это длина массива n , а следующие n чисел – введенные вручную элементы массива. Далее в зависимости от введенных данных и вызываемого метода выводится результат работы (элементы массива через пробел, текст ошибки или число).

A terminal window with a dark background and light gray text. The title bar shows three dots and the word "terminal". The output consists of four lines: the number 7, followed by the sequence 1 2 3 5 8 13 21, then the number 2, and finally the sequence 1 2 3 8 13 21.

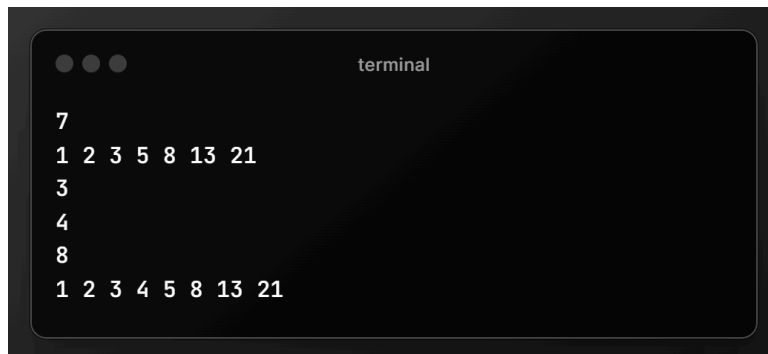
```
terminal
7
1 2 3 5 8 13 21
2
1 2 3 8 13 21
```

Рисунок 1 – Тест №1

A terminal window with a dark background and light gray text. The title bar shows three dots and the word "terminal". The output consists of four lines: the number 2, followed by the sequence 9 11, then the number 2, and finally the error message "Невозможно удалить элемент: неверный индекс!".

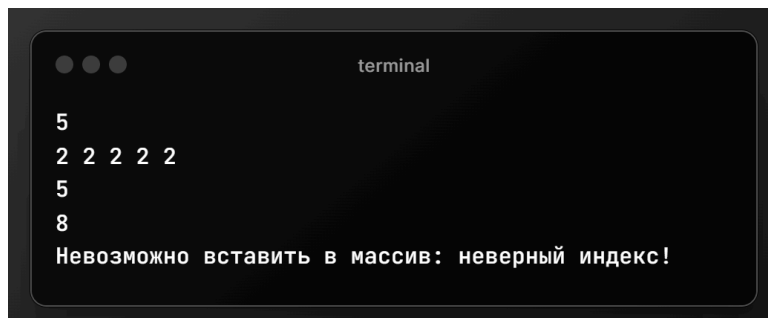
```
terminal
2
9 11
2
Невозможно удалить элемент: неверный индекс!
```

Рисунок 2 – Тест №2

A terminal window with a dark background and light gray text. The title bar shows three dots and the word "terminal". The output consists of seven lines: the number 7, followed by the sequence 1 2 3 5 8 13 21, then the numbers 3, 4, and 8, and finally the sequence 1 2 3 4 5 8 13 21.

```
terminal
7
1 2 3 5 8 13 21
3
4
8
1 2 3 4 5 8 13 21
```

Рисунок 3 – Тест №3

A terminal window with a dark background and light gray text. The title bar shows three dots and the word "terminal". The output consists of five lines: the number 5, followed by the sequence 2 2 2 2 2, then the numbers 5 and 8, and finally the error message "Невозможно вставить в массив: неверный индекс!".

```
terminal
5
2 2 2 2 2
5
8
Невозможно вставить в массив: неверный индекс!
```

Рисунок 4 – Тест №4

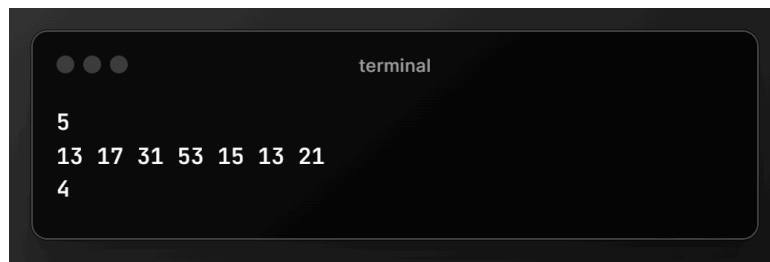


Рисунок 5 – Тест №5

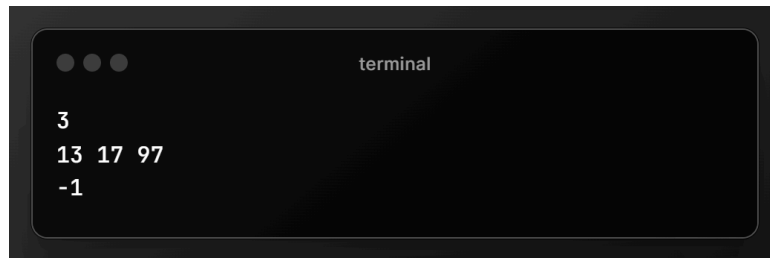


Рисунок 6 – Тест №6