

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 №1**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. Абстрактный тип данных задачи и его реализация на одномерном статическом массиве

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы ИКБО-06-22 |  | Кликушин В.И. |
| Принял старший преподаватель |  | Скворцова Л.А. |

Москва 2023

ОГЛАВЛЕНИЕ

[1 УСЛОВИЕ ЗАДАЧИ И ЗАДАНИЕ ВАРИАНТА 3](#_Toc128404228)

[2 АБСТРАКТНЫЙ ТИП ДАННЫХ ДЛЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 4](#_Toc128404229)

[3 РАЗРАБОТКА ПРОГРАММЫ 6](#_Toc128404230)

[3.1 РЕАЛИЗАЦИЯ ДАННЫХ АТД 6](#_Toc128404231)

[3.2 АЛГОРИТМЫ РЕАЛИЗОВАННЫХ ОПЕРАЦИЙ 6](#_Toc128404232)

[3.2.1 Удаление элемента в заданной позиции 6](#_Toc128404233)

[3.2.2 Вставка элемента в заданную позицию 6](#_Toc128404234)

[3.2.3 Вставка нового элемента в массив после элемента – палиндрома 6](#_Toc128404235)

[3.3 ТАБЛИЦЫ ТЕСТОВ ДЛЯ ОПЕРАЦИЙ ВАРИАНТА 7](#_Toc128404236)

[3.3.1 Тест поиска элемента - палиндрома 7](#_Toc128404237)

[3.3.2 Тест на удаление элементов массива, стоящих перед палиндромом 7](#_Toc128404238)

[3.3.3 Тест на вставку нового элемента после элемента - палиндрома 7](#_Toc128404239)

[3.4 КОД ПРОЕКТА 8](#_Toc128404240)

[3.4.1 Код файла TypeX.h 8](#_Toc128404241)

[3.4.2 Код файла TypeX.cpp 8](#_Toc128404242)

[3.4.3 Код файла siaod\_1.cpp 11](#_Toc128404243)

[3.5 СКРИНЫ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ 14](#_Toc128404244)

[4 ВЫВОДЫ 20](#_Toc128404245)

[5 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ 21](#_Toc128404246)

# 1 УСЛОВИЕ ЗАДАЧИ И ЗАДАНИЕ ВАРИАНТА

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

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.

Операции над множеством, которые требуется реализовать:

1. заполнение структуры данных значениями (с клавиатуры, применение датчика случайных чисел);
2. вывод структуры в консоль;
3. удалить элемент в заданной позиции;
4. вставить элемент в заданную позицию;
5. найти позицию элемента массива (первое вхождение), которое является палиндромом;
6. удалить элементы массива, расположенные непосредственно перед элементом, содержащим число палиндром;
7. вставить новый элемент в массив после элемента массива, который является палиндромом.

# 2 АБСТРАКТНЫЙ ТИП ДАННЫХ ДЛЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

АТД TypeX

{

Данные

n – количество элементов множества

data – список значений элементов множества, реализованный на статическом массиве

pos – позиция элемента, который является палиндромом

Операции

1. заполнение структуры данных значениями с клавиатуры

Предусловие. Ссылка на множество типа TypeX

Постусловие. Заполненный массив data из n элементов. Нет возвращаемого значения.

**input\_data(TypeX& data);**

1. заполнение структуры данных значениями – применение датчика случайных чисел

Предусловие. Ссылка на множество типа TypeX

Постусловие. Заполненный массив data из n случайных элементов. Нет возвращаемого значения.

**Random\_input\_data(TypeX& data);**

1. вывод структуры в консоль

Предусловие. Ссылка на множество типа TypeX

Постусловие. Вывод значений из массива. Нет возвращаемого значения.

**output\_data(TypeX& data);**

1. удалить элемент в заданной позиции

Предусловие. Ссылка на множество типа TypeX. Передача значения переменной pos в функцию.

Постусловие. Массив data с (n-1) элементами и удалённым элементов в позиции pos. Нет возвращаемого значения.

**deleteElem(TypeX& data, int pos);**

1. вставить элемент в заданную позицию

Предусловие. Ссылка на множество типа TypeX. Передача вставляемого элемент типа typeitem по значению. Передача позиции вставляемого элемента по значению.

Постусловие. Массив data с (n + 1) элементом и вставленным элементом n в pos позицию. Нет возвращаемого значения.

**insertElem(TypeX& data, typeitem n, int pos);**

*Дополнительные операции*

1. найти позицию элемента массива (первое вхождение), которое является палиндромом

Предусловие. Ссылка на множество типа TypeX.

Постусловие. Вывод на экран позиции элемента – палиндрома. Если такового не найдено, вывод сообщения: «Палиндрома не найдено!».

**special\_findElem(TypeX& data);**

1. удалить элементы массива, расположенные непосредственно перед элементом, содержащим число палиндром

Предусловие. Ссылка на множество типа TypeX.

Постусловие. Массив data с некоторым количеством элементов и удалёнными элементами, стоящими до Pos позиции. Нет возвращаемого значения.

**special\_deleteElem(TypeX& data);**

1. вставить новый элемент в массив после элемента массива, который является палиндромом

Предусловие. Ссылка на множество типа TypeX.

Постусловие. Массив data с (n+1) элементов и вставленным элементом n типа typeitem, если найдена позиция элемента – палиндрома, в противном случае – исходный массив data с n элементами.

**special\_insertElem(TypeX& data, typeitem n);**

}

# 3 РАЗРАБОТКА ПРОГРАММЫ

## 3.1 РЕАЛИЗАЦИЯ ДАННЫХ АТД

const unsigned int N = 100;

struct TypeX

{

unsigned int n = 0;

unsigned int Pos = N + 1;

typeitem X[N];

TypeX();

};

void input\_data(TypeX& data);

void Random\_input\_data(TypeX& data);

void output\_data(TypeX& data);

void deleteElem(TypeX& data, int pos);

void insertElem(TypeX& data, typeitem n, int pos);

bool check\_palindrom(typeitem n);

int special\_findElem(TypeX& data);

void special\_deleteElem(TypeX& data);

void special\_insertElem(TypeX& data, typeitem n);

## 3.2 АЛГОРИТМЫ РЕАЛИЗОВАННЫХ ОПЕРАЦИЙ

## 3.2.1 Удаление элемента в заданной позиции

если pos < 0 или pos >= n

то вывести «Неверная позиция»

иначе

для i от (pos + 1) до (n – 1)

X[i - 1] = X[i]

n = n – 1

## 3.2.2 Вставка элемента в заданную позицию

если pos < 0 или pos > n или pos == N

то вывести «Невозможно вставить элемент»

иначе

для i от n до (pos + 1)

X[i] = X[i - 1]

X[pos] = x

n = n + 1

## 3.2.3 Вставка нового элемента в массив после элемента – палиндрома

## 3.3 ТАБЛИЦЫ ТЕСТОВ ДЛЯ ОПЕРАЦИЙ ВАРИАНТА

## 3.3.1 Тест поиска элемента - палиндрома

|  |  |  |
| --- | --- | --- |
| Найти позицию элемента массива (первое вхождение), которое является палиндромом | | |
| Номер теста | Входные данные | Эталон результата |
| 1 | N = 100, n = 4  arr = {1, 23, 2424, 10} | Позиция элемента: 0  Pos = 0  Код работы программы 0 |
| 2 | N = 100, n = 5  arr = {17, 18, 31, 25, 99} | Позиция элемента: 4  Pos = 4  Код работы программы 0 |
| 3 | N = 100, n = 3  arr = {13, 1234321, 74} | Позиция элемента: 1  Pos = 1  Код работы программы 0 |
| 4 | N = 100, n = 4  arr = {12, 16, 145, 28} | Элемент не найден  Код работы программы 1 |

## 3.3.2 Тест на удаление элементов массива, стоящих перед палиндромом

|  |  |  |
| --- | --- | --- |
| Удалить элементы массива, расположенные непосредственно перед элементом, содержащим число палиндром | | |
| Номер теста | Входные данные | Эталон результата |
| 1 | N = 100, n = 4  arr = {1, 15, 28, 433} | N = 100, n = 4  arr = {1, 15, 28, 433}  Код работы программы 0 |
| 2 | N = 100, n = 5  arr = {15, 423, 221, 76, 909} | N = 100, n = 1  arr = {909}  Код работы программы 0 |
| 3 | N = 100, n =5  arr = {987, 13, 121, 7, 5} | N = 100, n = 3  arr = {121, 7, 5}  Код работы программы 0 |
| 4 | N = 100, n = 4  arr = {13, 75, 543, 10} | N = 100, n =4  arr = {13, 75, 543, 10}  Код работы программы 0 |

## 3.3.3 Тест на вставку нового элемента после элемента - палиндрома

|  |  |  |
| --- | --- | --- |
| Вставить новый элемент в массив после элемента массива, который является палиндромом | | |
| Номер теста | Входные данные | Эталон результата |
| 1 | N = 100, x = 23, n = 3  arr = {323, 1, 2} | N = 100, n = 4  arr = {323, 23, 1, 2}  Код работы программы 0 |
| 2 | N = 100, x = 1, n = 3  arr = {17, 165, 777} | N = 100, n = 4  arr = {17, 165, 777, 1}  Код работы программы 0 |
| 3 | N = 100, x = 17, n = 4  arr = {17, 10, 11, 10} | N = 100, n = 5  arr = {17, 10, 11, 17, 10}  Код работы программы 0 |
| 4 | N = 100, x = 512, n = 3  arr = {75, 72, 25} | Элемент – палиндром не найден  Код работы программы 1 |

## 3.4 КОД ПРОЕКТА

## 3.4.1 Код файла TypeX.h

#pragma once

#include <iostream>

using namespace std;

typedef int typeitem;

const unsigned int N = 100;

struct TypeX

{

unsigned int n = 0;

unsigned int Pos = N + 1;

typeitem X[N];

TypeX();

};

void input\_data(TypeX& data);

void Random\_input\_data(TypeX& data);

void output\_data(TypeX& data);

void deleteElem(TypeX& data, int pos);

void insertElem(TypeX& data, typeitem n, int pos);

bool check\_palindrom(typeitem n);

int special\_findElem(TypeX& data);

void special\_deleteElem(TypeX& data);

void special\_insertElem(TypeX& data, typeitem n);

## 3.4.2 Код файла TypeX.cpp

#include "TypeX.h"

TypeX::TypeX()

{

n = 0;

}

void input\_data(TypeX& data)

{

cout << "Введите " << data.n <<" "<< "чисел: ";

for (int i = 0; i < data.n; i++)

{

cin >> data.X[i];

}

}

void Random\_input\_data(TypeX& data)

{

srand(time(NULL));

for (int i = 0; i < data.n; i++)

{

data.X[i] = rand() % 100;

}

}

void output\_data(TypeX& data)

{

cout << "Data is " << data.n << " numbers: ";

for (int i = 0; i < data.n; i++)

{

cout << data.X[i] << " ";

}

cout << endl;

}

void deleteElem(TypeX& data, int pos)

{

if (pos >= 0 and pos < data.n)

{

for (int i = pos+1; i < data.n; i++)

{

data.X[i - 1] = data.X[i];

}

data.n--;

}

else

{

cout << "Ошибка в позиции" << endl;

}

}

void insertElem(TypeX& data, typeitem n, int pos)

{

if (pos >= 0 and pos <= data.n)

{

for (int i = data.n; i > pos; i--)

{

data.X[i] = data.X[i - 1];

}

data.X[pos] = n;

data.n++;

}

else

{

cout << "Ошибка в позиции" << endl;

}

}

bool check\_palindrom(typeitem n)

{

int copy\_n = n, last\_digit,res = 0;

while (n > 0)

{

last\_digit = n % 10;

res = (res \* 10) + last\_digit;

n /= 10;

}

return res == copy\_n;

}

int special\_findElem(TypeX& data)

{

for (int i = 0; i < data.n; i++)

{

if (check\_palindrom(data.X[i]))

{

data.Pos = i;

return 0;

}

}

return -1;

}

void special\_deleteElem(TypeX& data)

{

while (data.Pos != 0) { deleteElem(data, --data.Pos); }

}

void special\_insertElem(TypeX& data,typeitem n)

{

insertElem(data, n, data.Pos + 1);

}

## 3.4.3 Код файла siaod\_1.cpp

#include <iostream>

using namespace std;

#include "TypeX.h"

#include <ctime>

int main()

{

system("chcp 1251");

int n, pos;

int num, number;

typeitem x;

TypeX arr;

while (true)

{

cout << "-----------------------------------------------------------------------------------------------------------" << endl;

cout << " Меню\n";

cout << " 1. Заполнить множество с клавиатуры\n";

cout << " 2. Заполнить множество случайными числами\n";

cout << " 3. Вывести множество\n";

cout << " 4. Удалить элемент в позиции\n";

cout << " 5. Вставить элемент в позицию\n";

cout << " 6. Найти позицию элемента, который является палиндромом\n";

cout << " 7. Удалить элементы массива, расположенные непосредственно перед элементом, содержащим число палиндром\n";

cout << " 8. Вставить новый элемент в массив после элемента массива, который является палиндромом\n";

cout << " 9. Завершить работу\n";

cin >> num;

switch (num)

{

case 1:

cout << " Введите размер множества: ";

cin >> n;

arr.n = n;

if (n == 0 or n > N)

{

cout << "Неккоректный размер множества" << endl;

}

else

{

input\_data(arr);

}

break;

case 2:

cout << " Введите размер множества: ";

cin >> n;

arr.n = n;

if (n == 0 or n > N)

{

cout << "Неккоректный размер множества" << endl;

}

else

{

Random\_input\_data(arr);

}

break;

case 3:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else

{

output\_data(arr);

}

break;

case 4:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else

{

cout << " Введите позицию удаляемого элемента: " << endl;

cin >> pos;

deleteElem(arr, pos);

}

break;

case 5:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else

{

cout << " введите элемент для вставки: " << endl;

cin >> x;

cout << " Введите позицию вставляемого элемента: " << endl;

cin >> pos;

insertElem(arr, x, pos);

}

break;

case 6:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else

{

if (special\_findElem(arr) != -1)

{

cout << "Позиция элемента палиндрома: " << arr.Pos << endl;

}

else

{

cout << "Элемента палиндрома не найдено!" << endl;

}

}

break;

case 7:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else if (special\_findElem(arr)==-1)

{

cout << " элемента - палиндрома не найдено! " << endl;

}

else

{

special\_deleteElem(arr);

}

break;

case 8:

if (arr.n == 0)

{

cout << " пустое множество" << endl;

}

else if (special\_findElem(arr)==-1)

{

cout << " элемента - палиндрома не найдено! " << endl;

}

else

{

cout << " Введите элемент для вставки: " << endl;

cin >> x;

special\_insertElem(arr, x);

}

break;

case 9:

return 0;

default: cout << " Нет такого пункта" << endl;

}

}

return 0;

}

## 3.5 СКРИНЫ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Результат теста 1.1

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Результат теста 1.2

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 3 – Результат теста 1.3

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 4 – Результат теста 1.4

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 - Результат теста 2.1

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 6 – Результат теста 2.2

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 7 – Результат теста 2.3

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 8 – Результат теста 2.4

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 9 – Результат теста 3.1

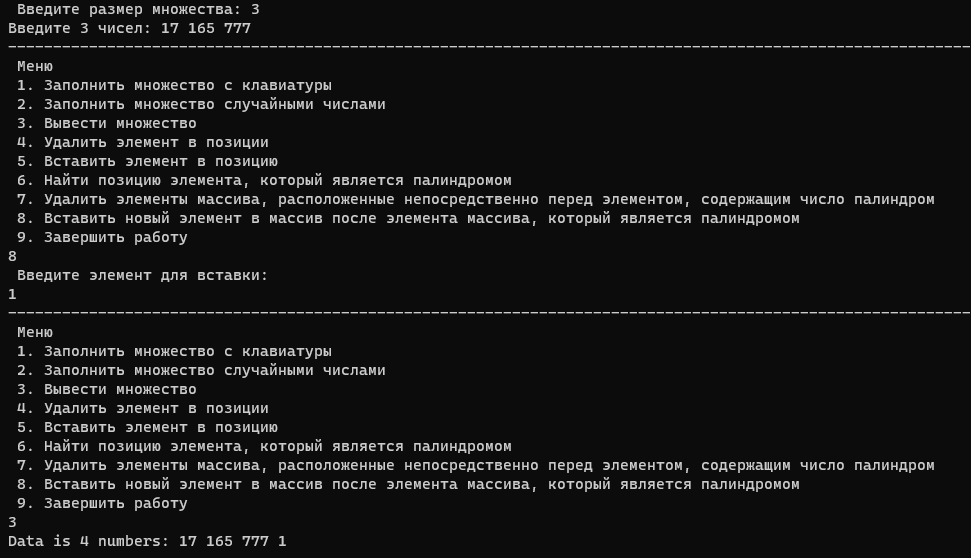


Рисунок 10 – Результат теста 3.2

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 11 – Результат теста 3.3

Изображение выглядит как текст

Автоматически созданное описаниеРисунок 12 – Результат теста 3.4

# 4 ВЫВОДЫ

В ходе выполнения практической работы были получены навыки создания АТД для поставленной задачи и реализации функций для соответствующей задачи. Была изучена структура C++ для реализации абстрактного типа данных.

Оценивая реализацию сложной структуры данных из примеров 1 и 2 можно отметить, что реализация в примере один получается более компактная и удобная для восприятия программиста за счет использования struct, однако на время выполнения и конечный результат выбор способа реализации АТД не влияет.

# 5 ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Учебно-методическое пособие СиАОД (часть 1)
2. Приложение к практическим работам – СДО (online-edu.mirea.ru)