

Самостоятельная работа 2

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

Стандартные структуры данных языка программирования для представление многоэлементных однородных структур данных задачи в программе. Одномерный статический массив.

Цель:

- освоение технологии разработки программ на основе абстрактного типа;
- приобретение умений и навыков определения и реализации абстрактного типа данных задачи на наиболее эффективной структуре представления данных в программе;
- приобретение навыков по реализации алгоритмов операций над массивом через аппарат функций языка C++;
- приобретение навыков реализации операций модификации (вставка и удаление элементов) одномерного массива и других операций обработки массива как структуры данных.

1. Введение в АТД

Разработка программной реализации задачи предусматривает определение абстрактного типа данных (далее АТД) задачи. Абстракция определяет область и структуру данных вместе с набором операций, которые требуются для доступа к данным при решении задачи. АТД – это тип данных, разработанный пользователем. Для каждой задачи разрабатывается свой АТД. АТД является независимым от реализации (языка программирования, структур данных языка программирования) и позволяет программисту сосредоточиться на идеализированных моделях данных и операциях над ними.

При программной реализации АТД программист выбирает такую структуру представления данных, чтобы алгоритмы операций были наиболее эффективными.

В данной практической работе будет определен АТД задачи, который должен быть реализован на статическом одномерном массиве. В набор операций АТД включаются: операции ввода и вывода массива; операции,

определенные в индивидуальном варианте. В следующей работе будут использованы другие структуры реализации этой АТД.

Формат АТД.

```
АТД Имя_АТД
{
    Данные:
        Описание структуры данных (свойства структуры), типов.
    Операции:
        _Операция 1 (указать: действие выполняемое с данными, предусловие,
        постусловие, заголовок (Имя операции и список входных данных в круглых
        скобках).
        Операция 2.
        .....
        Операция k.
}
```

Пример объявления операции в АТД.

```
//Операция: удаление из списка значений А заданного значения х.
//Предусловие: А – входные данные, n>0.
//Постусловие: результат при успешном поиске – pos, при безуспешном код
завершения -1.
    searchXinA(Имя_АТД, pos);.- заголовок
```

2. Требования к выполнению практической работы

Выполнить одно задание, в соответствии с условием задания. Задачи для задания представлены в табл. 6.

Применить технологию разработки программы на основе абстрактного типа данных.

3. Задание

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.

4. Требования к выполнению задания

1. Разработать абстрактный тип данных задачи вашего варианта, используя шаблон:

АТД *Имя_АТД*

{

Данные (описание свойств структуры данных задачи).

n – количество элементов множества.

A – список значений элементов множества.

pos – позиция элемента.

Примечание. Имена данных могут быть другими.

Операции (объявления операций).

Общие для всех вариантов.

- 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.

Дополнительные операции.

Операции, определенные в варианте и выявленные разработчиком как вспомогательные, для улучшения структуризации кода операции.

}

Реализовать список *A* из АТД можно на трех видах структур данных: статический массив, динамический массив, *vector*. В задании 1 рассмотрим реализацию на статическом массиве.

2. Реализовать АТД задачи, используя для представления списка значений множества *A* статический массив максимального размера *N*.

- 2.1. Выполнить реализацию множества, используя тип данных *struct* языка C++, включив поля: статический массив для хранения значений множества, поле (например, с именем *n*) для хранения текущего размера множества, поле (например, с именем *N*) для хранения размера статического массива. При этом $n \leq N$.

- 2.2. Разработать и записать на псевдокоде алгоритмы операций:

- 1) удалить элемент в заданной позиции ($pos \leq n$);

2) вставить элемент в заданную позицию ($pos \leq n$).

2.3. Реализовать все операции, заявленные в АД, на реализованной структуре представления данных. Каждая операция оформляется отдельной функцией с параметрами.

Примечание. Реализацию АД можно выполнить в отдельном файле заголовка, разрабатываемого проекта. В основной программе его нужно будет подключить.

3. Разработать наборы тестов для тестирования дополнительных операций.

Набор тестов для каждой операции оформить отдельной таблицей по формату, представленному в табл. 4.

Таблица 4. Шаблон таблицы тестов

Название алгоритма операции		
Номер теста	Входные данные	Эталон результата (ожидаемый результат)

Пример оформления теста для операции: из массива *удалить элемент в заданной позиции pos* представлен в табл. 5. Реализация массива на C++. Алгоритм возвращает код завершения 0 – успешно, 1 не успешно.

Таблица 5. Пример оформления таблицы тестов

Удалить элемент в заданной позиции <code>int int delPosInA(type *A, int &n, int pos)</code>		
Номер теста	Входные данные	Эталон результата
1.	N=100, n=10, pos=2 A {1,2,3,4,5,6,7,8,9,0,.....}	Эталон результата N=100, n=9, A {1,2, 4,5,6,7,8,9,0} Код завершения 0.
2.	N=100, n=10, pos=1 A {1,2,3,4,5,6,7,8,9,0,.....}	N=100, n=9 A {1,3,4,5,6,7,8,9,0} Код завершения 0.
3.	N=100, n=5, pos=4 A {1,2,3,4,5,.....}	N=100, n=5, pos=4 A {1,2,3,4} Код завершения 0.
4.	N=100, n=5, pos=10 A {1,2,3,4,5,.....}	Код завершения 1: нет заданной позиции

4. Разработать основную программу, демонстрирующую выполнение всех операций над структурой данных на подготовленных тестах. Программа должна управлять всем процессом посредством текстового меню.

5. Выполнить тестирование основной программы на подготовленных тестах.

6. Оформить отчет.

5. Варианты задач

Таблица 6. Варианты задач к заданию практической работы 1

№	Дополнительные операции над элементами структуры
1.	<ol style="list-style-type: none">1. Найти позицию элемента массива, являющегося простым числом*.2. Вставить новый элемент в массив в позицию, следующую за первым простым числом в массиве.3. Удалить каждый элемент массива, который кратен 7.
2.	<ol style="list-style-type: none">1. Вставить новое значение в массив перед первым элементом массива. *2. Определить, образуют ли числа массива арифметическую прогрессию.3. Удалить элементы массива, в значениях которых первая и последняя цифры одинаковы.
3.	<ol style="list-style-type: none">1. Вставить новый элемент в массив перед элементом в заданной позиции.2. Найти последнее вхождение в массив числа, у которого равны первая и последняя цифры.3. Удалить элементы массива кратные 5, заменив его значением последнего элемента массива. *
4.	<ol style="list-style-type: none">1. Вставить новое значение после значения в заданной позиции.2. Определить, сколько раз входит в массив максимальное значение массива (одним алгоритмом) *.3. Удалить все числа массива, которые являются совершенными числами (число равно сумме своих делителей кроме самого числа: 6, 28).
5.	<ol style="list-style-type: none">1. Вставить новый элемент в массив перед элементом, у которого четное количество цифр.2. Удалить все четные числа массива.3. Найти максимальное число среди элементов массива, расположенных на четных местах. *
6.	<ol style="list-style-type: none">1. Найти позицию элемента массива цифры которого упорядочены по возрастанию. Считать, что такое число одно.2. Вставить новый элемент после элемента, цифры которого упорядочены по возрастанию.3. Удалить число, которое расположено перед числом, цифры которого упорядочены по возрастанию.
7.	<ol style="list-style-type: none">1. Найти позицию элемента массива, цифровой корень которого равен 7. Подсказка. Цифровой корень – это однозначное число. Алгоритм определения цифрового корня: дано число 277, сумма его цифр 16 – двухзначное; снова сумма но уже 16 равна 7 – уже однозначное – это цифровой корень числа 277. *2. Вставить новый элемент перед элементом, цифровой корень которого равен 7. Считать, что такое число одно.3. Удалить элементы массива цифровой корень которых равен 7.

№	Дополнительные операции над элементами структуры
8.	<p>1. Найти позицию элемента массива, наибольшая цифра значения которого – это первая цифра числа. *</p> <p>2. Вставить новый элемент в массив перед элементом, наибольшая цифра значения которого – это первая цифра числа.</p> <p>3. Удалить элементы массива, наибольшая цифра значения которых – это первая цифра числа.</p>
9.	<p>1. Найти позицию элемента массива (первое вхождение), которое является палиндромом.</p> <p>2. Удалить элементы массива, расположенные непосредственно перед элементом, содержащим число палиндром.</p> <p>3. Вставить новый элемент в массив после элемента массива, который является палиндромом. *</p>
10.	<p>1. Найти позицию элемента массива (первое вхождение), которое является совершенным (число равно сумме своих делителей кроме самого числа: 6, 28). *</p> <p>2. Вставить новый элемент в массив после элемента, который является совершенным.</p> <p>3. Удалить элемент массива, расположенный перед элементом, содержащим совершенное число.</p>
11.	<p>1. Найти позицию элемента массива, цифры которого (слева направо) образуют последовательность Фибоначчи. *</p> <p>2. Вставить новый элемент в массив после элемента, цифры которого образуют последовательность чисел Фибоначчи.</p> <p>3. Удалить элемент массива, расположенный перед элементом, цифры которого образуют последовательность чисел Фибоначчи.</p>
12.	<p>1. Найти позицию максимального элемента массива, среди четных чисел массива. *</p> <p>2. Вставить новый элемент в массив после элемента с максимальным значением среди четных чисел массива.</p> <p>3. Удалить элемент массива, расположенный перед элементом, с максимальным значением среди четных чисел массива.</p>
13.	<p>1. Найти максимальное значение массива.</p> <p>2. Вставить максимальное значение массива после элемента, у которого первая и последняя цифры равны. *</p> <p>3. Удалить элементы массива, цифры которых образуют последовательность чисел Фибоначчи, в которой первое и второе число равно 1.</p>
14.	<p>1. Найти позицию первого вхождения минимального значения среди отрицательных чисел массива. *</p> <p>2. Вставить новый элемент массива после минимального элемента массива.</p> <p>3. Удалить все элементы массива равные минимальному значению в массиве среди отрицательных чисел.</p>

№	Дополнительные операции над элементами структуры
15.	<p>1. Найти позицию элемента (первое вхождение) массива, у которого все цифры одинаковые. *</p> <p>2. Вставить новый элемент в массив после элемента, все у которого все цифры одинаковые.</p> <p>3. Удалить элементы, у которого все цифры одинаковые.</p>
16.	<p>1. Найти позицию элемента массива (первое вхождение) двоичный код значения которого содержит ровно три единицы.</p> <p>2. Вставить новый элемент в массив после элемента, двоичный код значения которого содержит ровно три единицы. *</p> <p>3. Удалить элементы массива, двоичный код значения которых содержит ровно три единицы.</p>
17.	<p>1. Найти позицию элемента массива, старшая цифра значения которого равна заданной.</p> <p>2. Вставить новый элемент в массив перед элементом массива, старшая цифра значения которого равна заданной.</p> <p>3. Удалить все элементы массива, старшая цифра значений которых равна заданной.</p>
18.	<p>1. Найти позицию элемента массива (первое вхождение) троичный код значения которого содержит ровно две двойки.</p> <p>2. Вставить новый элемент в массив после элемента, троичный код значения которого содержит ровно две двойки.</p> <p>3. Удалить элементы массива, троичный код значений которых содержит ровно две двойки. *</p>
19.	<p>1. Найти позицию элемента массива, значение которого содержит цифру 0.</p> <p>2. Вставить новый элемент в массив после элемента, значение которого не содержит цифру 0.</p> <p>3. Удалить элементы массива, значение которого содержит цифру 0.</p>
20.	<p>1. Найти позицию элемента массива, произведение цифр которого больше нуля и кратно трем. *</p> <p>2. Вставить новый элемент в массив перед элементом с максимальным значением.</p> <p>3. Удалить элемент массива, произведение цифр которого больше нуля и кратно трем.</p>
21.	<p>1. Найти позицию элемента массива, сумма цифр значения которого кратна 7.</p> <p>2. Вставить новый элемент в массив перед минимальным элементом, сумма цифр значения которого кратна 7.</p> <p>3. Удалить элементы массива, сумма цифр значения которого кратна 7.</p>
22.	<p>1. Найти позицию элемента массива значение которого делится на каждую из цифр числа.*</p> <p>2. Вставить в массив новый элемент после элемента, значение которого делится на каждую цифру значения.</p> <p>3. Удалить из массива все элементы, кратные трем.</p>

№	Дополнительные операции над элементами структуры
23.	<ol style="list-style-type: none"> 1. Определить, упорядочены ли значения в массиве по возрастанию. * 2. Если значения в массиве упорядочены по возрастанию, то удалить из массива элементы, которые кратны введенному значению. 3. Если значения в массиве не упорядочены по возрастанию, то вставить новый элемент в массив перед первым элементом.
24.	<ol style="list-style-type: none"> 1. Найти позиции (начальный и конечный) самой длинной, упорядоченной по возрастанию подпоследовательности (части массива). *<i>Считать подпоследовательностью элементы, следующие непосредственно, друг за другом.</i> 2. Вставить новый элемент перед элементом, с начальным индексом подпоследовательности. 3. Удалить все элементы найденной подпоследовательности.
25.	<ol style="list-style-type: none"> 1. Определить, сколько раз в массиве встречается максимальное значение и сформировать массив индексов этих элементов. * 2. Удалить все максимальные значения, используя массив их индексов. 3. Если в массиве только одно максимальное значение, то добавить такое же значение в массив.
26.	<ol style="list-style-type: none"> 1. Определить, упорядочены ли значения в массиве по возрастанию. 2. Если значения в массиве не упорядочены по возрастанию, то удалить из массива элементы, которые кратны введенному значению. 3. Если значения в массиве упорядочены по возрастанию, то вставить новый элемент в массив перед элементом с большим его по значению. *
27.	<ol style="list-style-type: none"> 1. Определить, упорядочены ли значения в массиве по убыванию. 2. Если значения в массиве не упорядочены по убыванию, то удалить из массива элементы, значения которых содержат цифру 5. 3. Если значения в массиве упорядочены по убыванию, то вставить новый элемент в массив перед элементом с меньшим его по значению. *.
28.	<ol style="list-style-type: none"> 1. Сформировать новый массив из простых чисел исходного массива, вставляя каждое значение (кроме первого значения) так, чтобы числа образовали в результате возрастающую последовательность. 2. Удалить минимальное число нового массива. 3. Определить у скольких чисел исходного массива количество делителей больше трех.
29.	<ol style="list-style-type: none"> 1. Сформировать новый массив из чисел исходного массива, сумма цифр которых кратна 7, вставляя каждое значение (кроме первого значения) так, чтобы числа образовали в результате убывающую последовательность. 2. Удалить минимальное число нового массива. 3. Определить у скольких чисел исходного массива цифры образуют возрастающую последовательность.

№	Дополнительные операции над элементами структуры
30.	<p>1. Определить сколько в массиве простых чисел Мерсенна. * <i>Считать натуральное число M простым числом Мерсенна, если оно удовлетворяет свойствам: 1) M – простое число 2) число $M+1$ является степенью двойки. Например, число $M=31$.</i></p> <p>2. Удалить минимальное число, которое является степенью числа 2.</p> <p>3. Вставить в массив новый элемент после элемента значение которого является максимальным простым числом Мерсенна.</p>

6. Структура отчета

Титульный лист.

Оглавление.

1. Условие задачи и задание варианта.

2. АТД задачи.

Включить текст АТД.

3. Разработка программы.

3.1. Включить реализацию данных АТД (представить код структуры).

3.2. Включить алгоритмы операций, указанных в пункте 2 требований к выполнению задания и задачи, отмеченной символом * в варианте.

3.3. Таблица тестов тестирования операций варианта.

3.4. Код проекта, включающий:

- файл заголовка, в котором представлена реализация АТД и его функций;
- основная программа.

4. Скрины результатов тестирования.

5. Используемые источники.

6. Выводы.

Указать какие знания умения и навыки вы получили в результате выполнения практической работы.

1. Дайте определение структуре данных.

2. Какие три операции над структурой данных называют базовыми?

3. Что определяет тип данных в языках программирования?

4. Сколько ячеек памяти связано с именем переменной простого типа?

5. Сколько ячеек памяти связано с именем массива размером n ?

6. Перечислите свойства, характеризующие структуру данных массив.

7. Какой тип языка программирования C++ пользуется для представления структуры данных, элементы которой различного типа?

8. Могут ли элементом структуры `struct` быть другие структуры: массив, строка, указатель на другую структуру, другая структура?

9. Определите тип данных для представления в программе точки на плоскости (декартова система координат). Определите структуру для хранения данных по n точкам.
10. Перечислите свойства структуры, организованной на типе struct.
11. Как называется область оперативной памяти, предназначенная для хранения значения простого типа?
12. Что определяет размер ячейки оперативной памяти?
13. Как можно описать организацию оперативной памяти на логическом уровне?
14. Что определяет абсолютный адрес ячейки памяти?
15. Какой способ хранения элементов структуры данных реализует одномерный массив?
16. Какой способ хранения элементов структуры данных реализуется при определении пользовательского типа struct в C++?
17. Что нужно знать (определить), чтобы получить прямой доступ к значению ячейки оперативной памяти?
18. Какой способ хранения структур данных позволяет обеспечить прямой доступ к элементам структуры?
19. Приведите формулу, по которой определяется адрес i - ого элемента массива x. Какой параметр в формуле представляет смещение и почему его так называют?
20. Что представляет собой структура данных, имеющая векторное хранение элементов? Опишите.
21. Приведите характеристики статического массива.
22. Какие переменные называют локальными?
23. Как организована область оперативной памяти для хранения локальных переменных?
24. Что определяет понятие «время жизни переменной»?
25. Какой оператор нужно включить в код функции, если алгоритм функции вычисляет одно значение простого и ссылочного типа (приведите пример)?
26. Какую ошибку допустил программист при разработке функции func?

```
#include <iostream>
const int n=10;
int *func(){
int a[n]={ 1,2,3,4,5,6,7,8,9,0};
return a;
}
```

```

int main(){
int *b=func();
for(int i=0;i<n;i++)
cout<<b[i]<< " ";
}

```

27. Статическому массиву А выделена память под 100 элементов. В него ввели только десять значений (n=10). Алгоритм функции func1 добавил по индексу 11 еще одно значение. Определите ошибку, которую допустил программист при создании функции func1?

```

#include <iostream>
const int m=100;
void func1(int *a,int n,int x){
a[n]=x; n++;
}
int main(){
int n=10;
int a[m]={1,2,3,4,5,6,7,8,9,0};
func1(a,n,20);
for(int i=0;i<n;i++)
cout<<a[i]<<"";
}

```

28. Как объявить типизированный указатель в программе на языке C++?
29. Как объявить указатель без типа в программе на языке C++?
30. Какие арифметические операции допустимы над указателем в C++?
31. Программист написал неверный код копирования массива, который содержал операторы:
 int x[]={1,2,3,4,5}; int *y=x;
 Объясните его ошибку.
32. Объясните суть операции «разыменование» для указателя.
33. Приведите модель выполнения операции удаления значения массива из заданной позиции с сохранением порядка следования значений после удаленного значения.
34. Приведите алгоритм удаления значения массива из заданной позиции, не сохраняя порядок следования значений после удаленного значения.
35. Укажите условие, при котором допустима операция вставки нового значения в статический массив.
36. Приведите модель выполнения операции вставки нового значения в заданную позицию массива с сохранением порядка следования значений после вставленного значения.

37. Опишите различие алгоритмов операций вставки и добавления нового значения в массив.