Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им.И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Алгоритмы и структуры данных»

Лабораторная работа №2

Тема: «Указатели»

Выполнил: студент группы

ПИ(б)-2-19 Улан уулу Нурдин

Проверила: Валеева А. А.

Бишкек – 2020

**I.Ответить на контрольные вопросы**

1. **Чем отличаются статические и динамические величины?**

**Статическими величинами** называются такие, память под которые выделяется во время компиляции и сохраняется в течение всей работы программы.

**Динамическими величинами** называются такие, память под которые отводится во время выполнения программы.

1. **Какая память называется динамически распределяемой?**

**Динамическое распределение памяти** — способ выделения [оперативной памяти](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) для объектов в программе, при котором выделение памяти под объект осуществляется во время выполнения [программы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0).

1. **Какой оператор используется для получения адреса переменной?**

Для получения адреса переменной используется оператор **“&”**.

1. **Какой оператор позволяет получить значение по адресу, содержащемуся в указателе?**

Для доступа к значению, сохраненному по адресу, содержащемуся в указателе, используется оператор **“\*”**.

1. **Что такое указатель?**

Указатели представляют собой объекты, значением которых служат адреса других объектов или функций. Как и ссылки, указатели применяются для косвенного доступа к объекту.

1. **Приведите примеры объявления указателей:**

1) int a;

int\* i = &a;

2) int p = 5;

int\* r = &p;

3) int a = 5;

int &p = a;

int\* r = &p;

1. **В чем различие между адресом, хранящимся в указателе, и значением, записанным по этому адресу?**

Адрес, сохраненный в указателе, - это адрес другой переменной.Значение, сохраненное по этому адресу, - это любое значение, сохраняемое в переменной, на которую ссылается указатель. Оператор **“\*”** возвращает значение, сохраненное по адресу, который хранится в указателе.

1. **Как выделить память под динамическую переменную? Как освободить память от динамической переменной?**

Для динамического выделения памяти одной переменной используется **оператор new:**

int\* a = new int;

Когда уже всё, что требовалось, выполнено с динамически выделенной переменной — нужно явно указать для С++ освободить эту память. Для переменных это выполняется с помощью **оператора** **delete**:

delete a;

1. **Что в языке C++ означает константа NULL?**

**NULL** в языке программирования C++ — макрос, объявленный в заголовочном файле stddef.h. Значением этого макроса является зависящая от реализации константа нулевого указателя. Константа нулевого указателя — это целочисленное константное выражение со значением 0.

1. **В чем различие между оператором разыменования и оператором получения адреса?**

Оператор разыменования **“\*”** возвращает значение, хранящееся по адресу, на который ссылается указатель. А оператор получения адреса **“&”** возвращает адрес переменной в памяти.

1. **В чем различие межу объявлениями**

***сonst int \*ptr1* и**

***int const ptr2***

Выражение **const int \* ptr1** объявляет, что переменная ptr1 представляет собой указатель на неизменяемое число типа **int**.

Выражение **int const ptr2** объявляет, что переменная ptr2 является постоянным указателем на некоторое число типа **int**.

1. **В каком случае возможно присваивание указателей?**

Указатель можно использовать в правой части оператора присваивания для присваивания его значения другому указателю. Если оба указателя имеют один и тот же тип, то выполняется простое присваивание, без преобразования типа.

1. **Что такое утечка памяти?**

**Утечка памяти** — процесс неконтролируемого уменьшения объёма свободной [оперативной](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) или [виртуальной](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) [памяти](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), связанный с ошибками в работающих [программах](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), вовремя не освобождающих ненужные участки памяти, или с ошибками системных служб контроля памяти.

1. **Какие ситуации приводят к возникновению в динамически распределяемой памяти "мусора"?**

Если динамическая величина теряет свой указатель, то она становится "мусором". В программировании под этим словом понимают информацию, которая занимает память, но уже не нужна.

1. **В чем разница между ссылкой и указателем?**

Указатель может быть переназначен любое количество раз, в то время как ссылка после привязки не может быть перемещена на другую ячейку памяти.

Указатели могут указывать "в никуда" (быть равными NULL), в то время как ссылка всегда указывает на определенный объект.

Нельзя получить адрес ссылки, как можно это сделать с указателями.

Не существует арифметики ссылок, в то время как существует арифметика указателей.

1. **Когда нужно использовать указатель, а не ссылку?**

Когда создаем динамический массив.

Если нужно узнать адрес переменной.

**Выполнить упражнения**

1. Объяснить смысл следующих объявлений переменных:

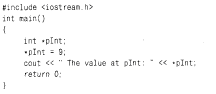
int \*Pone;

int Vtwo;

int \*PThree =&Vtwo;

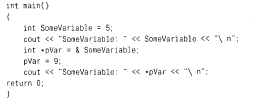
В переменной **Pone** нет смысла. Так как переменная **PThree** ссылаеться на переменную **Vtwo**.

2. Найти ошибку в следующем фрагменте:



**\*pInt = 9**; так нельзя инициализировать, указатель должен указывать на переменную, а не на число.

3. Найти ошибку в следующем фрагменте



Нельзя присвоить значению типа **int** сущность типа **int\***.

**Выполнить задания**

1. Написать программу для объявления переменной типа и указатель на этот тип. Сохранить адрес переменной в указателе. Используя указатель, присвоить переменной какое - либо значение.

int a = 10;

int\* b = &a;

cout << "Adress = " << b << "\tValue = " << \*b << endl;

1. Написать программу, которая объявляет переменную типа int, ссылку на значение типа int и указатель на значение типа int. Использовать указатель и ссылку для управления значением переменной типа int.

int a = 5;

cout << a << endl;

int\* b = &a;

cout << \*b << endl;

int& c = a;

cout << c << endl;

a = c + 1;

cout << a << endl;

a = \*b + 1;

cout << a << endl;

1. Написать программу, создающую блуждающий указатель.

int\* a = new int;

\*a = 5;

cout << \*a << endl;

delete a;

1. В упражнении 3 сделать блуждающий указатель нулевым.

int\* a = new int;

\*a = 5;

cout << \*a << endl;

delete a;

a = 0;

1. Написать программу, которая приводит к утечке памяти

int n; cin >> n;

int\* arr = new int[n];

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

arr[i] = i + 1;

cout << arr[i] << endl;}

**Выполнить индивидуальные задания по варианту: № 9**

**Определите, каким образом программа передает параметры (по значению или по ссылке) и как это влияет на результат:**

int &F39(int &n1, int &n2)

{

return n1 > n2 ? n1 : n2;

}

void main10()

{

int x=5, y=6;

F39(x,y) = 0;

F39(x,y) ++;

}

Передача параметров происходит через ссылку, и возвращается измененный результат.

**Разработать программу с функцией**

#include <iostream>

using namespace std;

void F(char\* arr, int n, char b)

{

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

{

if (arr[i] == b)

{

arr[i] = 0;

}

}

for (int j = 0; j < n; j++)

{

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

{

if (arr[i] == 0)

{

swap(arr[i], arr[i + 1]);

}

}

}

}

int main()

{

srand(time(NULL));

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

char m;

cin >> n;

char\* arr = new char[n];

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

{

cin >> arr[i];

}

cin >> m;

F(arr, n, m);

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

{

cout << arr[i];

}

delete[]arr;

return 0;

}