Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

Им. И. Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

ОТЧЕТ

Дисциплина «Алгоритмы и структуры данных»

Лабораторная работа № 2

Выполнил: студент группы

ПИ(б)-3-17 Инчин Никита

Проверила: Валеева А.

Бишкек – 2018

**Контрольные вопросы**

1. Чем отличаются статические и динамические величины?
2. Какая память называется динамически распределяемой?
3. Какой оператор используется для получения адреса переменной?
4. Какой оператор позволяет получить значение по адресу, содержащемуся в указателе?
5. Что такое указатель?
6. Приведите примеры объявления указателей
7. В чем различие между адресом хранящимся в указателе, и значением записанным по этому адресу?
8. Как выделить память под динамическую переменную? Как освободить память от динамической переменной?
9. Что в языке C++ означает константа NULL?
10. В чем различие между оператором разыменования и оператором получения адреса?
11. В чем различие межу объявлениями

*Const int \*ptr1* и

*Int\* const ptr2*

1. В каком случае возможно присваивание указателей?
2. Что такое утечка памяти?
3. Какие ситуации приводят к возникновению в динамически распределяемой памяти "мусора"?
4. В чем разница между ссылкой и указателем?
5. Когда нужно использовать указатель, а не ссылку?

**Ответы:**

1. Статические величины – когда переменная создаётся с фиксированным размером и пока выполняется программа её нельзя увеличить или уменьшить. Динамические величины – не имеет фиксированного размера и при выполнении программы возможно её увеличить или уменьшить. Высвободив занятую память и предложить другой размер.
2. Динамически распределяемой памятью – называется оперативная память компьютера или куча.
3. Для получения адреса переменной – используется ссылка (адрес a) &(амперсанд) a.
4. Оператор \*(звёздочка) – позволяет получить значение по адресу, определяет значение переменой, которое содержится по адресу в указателе.
5. Указатель – переменная, содержащая адрес объекта.
6. Объявления указателя: int a = 5; int \*p = &a; cout << \*p;
7. Адресом хранящимся в указателе – адрес другой переменой. Значение записанным по этому адресу - любое значение, записанное в переменой, на которую ссылается указатель. Оператор \* возвращает значение, записанное по адресу который хранится в указателе.
8. Для выделения памяти под динамическую переменную – используется оператор new (int \*a = new int). Для освобождения памяти – используется оператор delete a.
9. NULL – нулевой указатель – способ отметить указатели, которые очевидно не указывают на точный адрес в памяти.
10. Оператор разыменования (\*) – в возвращает значение, хранящееся по адресу, на который ссылается указатель. Оператор получения адреса (&) – возвращает адрес переменной в память.
11. Const int \*ptr1 – объявляет, что ptr1 указывает на постоянное целое число const int, а int \* const ptr2 – объявляет, что переменная ptr2 является постояным const указывает на целое число int.
12. Присваивание указателю:

* адреса существующего объекта;
* пустого значения;
* выделение участка динамической памяти и присваивание её адреса указателю;
* адреса области памяти в явном виде.

1. Утечка памяти – когда при выделении памяти под динамический массив мы не очищаем его, тем самым уменьшая объём свободной оперативной памяти.
2. Если динамическая величина теряет свой указатель, то она становится “мусором”.
3. Ссылка — это условное название, а указатель — это переменная, которая содержит адрес. Ссылки не могут быть нулевыми и не могут переназначаться.
4. Указатель нужно использовать:

* если в программе нужно назначить указателю новую переменную или если указатель нужно сделать нулевым.

**Выполнить упражнения**

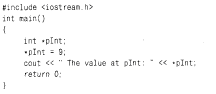
1. Объяснить смысл следующих объявлений переменных :

int \*Pone – объявляет указатель на int (целое значение).

int Vtwo – объявляет целочисленную переменную.

int \*PThree =&Vtwo - объявляет указатель на int, извлекает адрес переменой Vtwo и присваивает к указателю.

1. Найти ошибку в следующем фрагменте:

#include "iostream"

int main()

{

int x = 9;

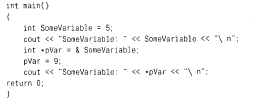
int \*pInt;

pInt = &x;

cout << \*pInt << endl;

return 0;

}

1. Найти ошибку в следующем фрагменте:

#include "iostream"

int main()

{

int SomeVarible = 5;

cout << "SomeVariable: " << SomeVarible << "\n";

int \*pVar = &SomeVarible;

SomeVarible = 9;

cout << "SomeVariable: " << \*pVar << "\n";

return 0;

}

**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.

**Постановка задачи:**

* Входные данные: переменная *(a)* типа short int и указатель *\*(p)* на этот тип.
* Выходные данные: присваиваем переменой (*a)* значение 100, используя указатель.

**Анализ:**

1. Объявить переменную типа и указатель на этот тип.
2. Сохранить адрес переменной в указателе.
3. Используя указатель, присвоить переменной какое - либо значение.

**Блок-схема:**

**Код:**

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\1.png#include "iostream"

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\11.pngusing namespace std;

void f(short int \*p);

int main()

{

short int i;

short int \*j;

j = &i;

f( j );

cout << i << endl;

system("pause");

return 0;

}

void f(short int \*p)

{

\*p = 100;

}

D:\Документы\Политех\2 курс\С++\ЛБ 2\скрины лб2\1.PNG

2.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.

**Постановка задачи:**

* Входные данные: переменная (х), ссылка (y = 10) и указатель (\*z = 6) типа int.
* Выходные данные: с начало используем ссылку переменная х = 10, после использования указателя переменная х = 6;

**Анализ:**

1. Объявить переменную, ссылку на значение и указатель на значение типа int.
2. Использовать указатель и ссылку для управления значением переменной типа int.

**Блок-схема:**

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\2.png**Код:**

#include <iostream>

using namespace std;

int main()

{

int x;

int& y = x;

int\* z = &x;

y = 10;

cout <<"X: "<< x << endl;

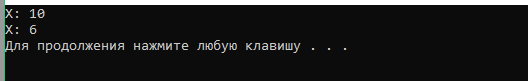
\*z = 6;

cout << "X: " << x << endl;

system("pause");

return 0;

}

****

3.Написать программу создающую блуждающий указатель

**Постановка задачи:**

* Входные данные: динамическая переменная (p) типа int;
* Выходные данные: блуждающий указатель;

**Анализ:**

1. Написать программу создающую блуждающий указатель.

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\3.png**Блок-схема:**

**Код:**

#include <iostream>

using namespace std;int main()

{

int \*p = new int;

delete p; // p становится блуждающим указателем

return 0;

}

4.В упражнении 3 сделать блуждающий указатель нулевым.

**Постановка задачи:**

* Входные данные: динамическая переменная (p) типа int;
* Выходные данные: блуждающий указатель становится нулевым.

**Анализ:**

Сделать блуждающий указатель нулевым.

**D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\4.pngБлок-схема:**

**Код:**

#include <iostream>

using namespace std;

int main() {

int \*p = new int;

delete p;

p = NULL; // p больше не блуждает

return 0;

}

5.Написать программу, которая приводит к утечке

**Постановка задачи:**

* Входные данные: динамическая переменная (p) типа int;
* Выходные данные: переменная цел-ого типа (p) и утечка памяти.

**Анализ:**

Написать программу, которая приводит к утечке.

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\51.png D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\5.png**Блок-схема:**

**Код:**

#include <iostream>

using namespace std;

int \* F();

int main()

{

int \*p = F();

cout << "P " << \*p << endl;

system("pause");

return 0;

}

int \* F()

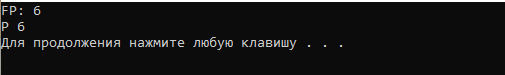
{

int \* p = new int(6);

cout << "FP: " << \*p << endl;

return p;

}



**Индивидуальное задание**

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

Задание 5)

//

int inc5(int &x) { x++; return x+1; }

 void main5 ()

 { int x,y,z; x = 5; y = inc5(x); z=inc5(x); z=inc5(z); }

//

Программа передает параметры по ссылке. Передача по ссылке позволяет возвратить из функции сразу несколько значений и использует сам объект, а не его значение.

Тестовый вариант:

Вход. Результат

1)x = 5 y = inc5(x)

x = 6; y = 7;

2)x = 6 z = inc5(x)

x = 7; z = 8;

3)x = 7 z = inc5(z)

z = 8 z = 9; z=10;

**Разработать программу с функцией:**

Задание 5.

Функция находит в строке пары одинаковых фрагментов и возвращает указатель на первый. С помощью функции найти все пары одинаковых фрагментов.

**Постановка задачи:**

* Входные данные: строка (s);
* Выходные данные: все пары одинаковых фрагментов.

**Анализ:**

* 1. Функция находит в строке пары одинаковых фрагментов.
  2. Функция возвращает указатель на первый.
  3. D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\61.pngС помощью ф-ции найти все пары одинаковых фрагментов.

**Блок-схема:**

**Код:**

#include <iostream>

#include <windows.h>

using namespace std;

char\* f(char\* s, char\*\* e) {

char\* p, \*i, \*j;

while (\*s != 0) {

p = s + 1;

while (\*p != 0) {

if (\*p == \*s && \*(p + 1) == \*(s + 1))

break;

p++; }

if ((p - s) > 1) {

i = s;

j = p;

while (\*i && (\*i == \*j)) {

j++;

i++; }

if (i == p) {

if (e != 0)

\*e = j;

return s; } }

D:\Документы\Политех\2 курс\С++\ЛБ 2\БС лб2\6.pngs++; }

return 0; }

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char\* p, \*e;

int size = 256;

char\* s = new char[size];

cin.getline(s, size);

p = f(s, 0);

cout << "\n"<< "Все пары: " << endl;

for (p = &s[0]; (p = f(p, &e)) != 0; p = e) {

while (p != e)

cout << \*p++;

cout << endl;

}

system("pause");

return 0;

}

