Algorithms and Data Structures

Laboratory work 2

Abakirov Nursultan

SE (English) 1-16

**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 - память под величины отводится во время выполнения программы. Раздел оперативной памяти, распределяемый статически, называется **статической памятью**; динамически распределяемый раздел памяти называется **динамической памятью**.

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?

Амперсант &

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

=

1. Что такое указатель?
2. Приведите примеры объявления указателей

int \*pointer = new int;

1. В чем различие между адресом хранящимся в указателе, и значением записанным по этому адресу?
2. Как выделить память под динамическую переменную? Как освободить память от динамической переменной?
3. Что в языке C++ означает константа NULL?
4. В чем различие между оператором разыменования и оператором получения адреса?
5. 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?

**Выполнить упражнения**

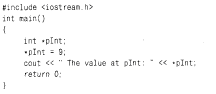
1. Объяснить смысл следующих объявлений переменных :

int \*Pone

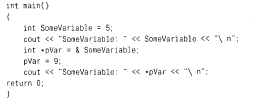
int Vtwo

int \*PThree =&Vtwo

2. Найти ошибку в следующем фрагменте:



4. Найти ошибку в следующем фрагменте



**Задания на выполнение**

#include <iostream>

#include <stdlib.h>

#include <time.h>

using namespace std;

int main(){

short task;

cin >> task;

if (task == 1){task1();}

else if (task == 2){task2();}

else if (task == 3){task3();}

else if(task == 4){task4();}

else if(task == 5){task5();}

else if(task == 6){task6();}

else{cout << "invalid input";}

}

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 task1(){

short a;

short \*pointer;

pointer = &a;

\*pointer = 5;

cout << "a = " << a << 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.

void task2(){

int x;

int \*pointer = &x;

int &linker = x;

\*pointer = 10;

cout << "pointer = " << x << endl;

linker = 5;

cout << "reference = " << x << endl;

}

1. Написать программу создающую блуждающий указатель

void task3(){

int x;

int \*pointer = new int;

pointer = &x;

\*pointer = 5;

cout << "before: " << x << endl;

delete pointer;

\*pointer = 10;

cout << "after: " << \*pointer << endl;

}

1. В упражнении 3 сделать блуждающий указатель нулевым

void task4(){

int x;

int \*pointer = new int;

pointer = &x;

\*pointer = 5;

cout << "before: " << x << endl;

delete pointer;

pointer = new int;

\*pointer = NULL;

cout << "after: " << pointer << endl;

}

1. Написать программу, которая приводит к утечке памяти

void task5(){

int \*a = new int;

\*a = 9;

cout << \*a << endl;}

1. Написать программу для создания динамического массива (A[10]), используя генератор случайных чисел.

void task6(){

srand(time(NULL));

short length;

cin >> length;

short \*a = new short[length];

for(short i = 0; i < length; i++){

a[i] = rand() % 1000 + 0;}

for(short i = 0; i < length; i++){

cout << a[i] << " ";}

cout << endl;

delete a;}