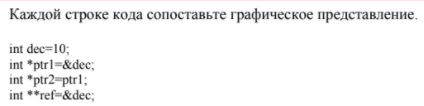
Самостоятельная работа No7

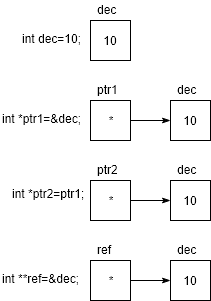
Указатели на простейшие типы.

Операции над указателями.

Задание 1.



Графическое представление:



Код:

#include <windows.h>

#include <stdio.h>

**int** main (**void**){

**int** dec=10;

printf("dec = %d**\n**", dec);

**int** \*ptr1=&dec;

printf("\*ptr1 = &dec = %x**\n**", ptr1);

**int** \*ptr2=ptr1;

printf("\*ptr2 = ptr1 = %x**\n**", ptr2);

**int** \*\*ref=&dec;

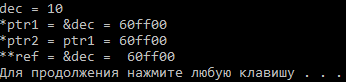
printf("\*\*ref = &dec = %x**\n**", ref);

system("pause");

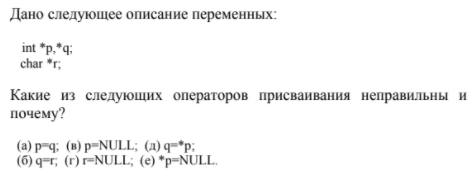
**return** 0;

}

Результат:



Задание 2.



Ответ:

(а) Вариант возможен.  
(б) Неверен, присвоение указателя другого типа приведёт к ошибке.

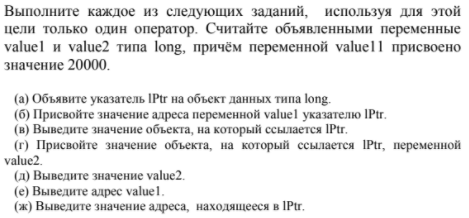
(в) Вариант возможен.

(г) Вариант возможен.

(д) Неверный вариант, т.к. происходит присвоение указателю объекта, содержащегося по адресу неинициализированного указателя.

(е) NULL определяет константу для нулевого указателя, здесь же наблюдается попытка присвоения NULL объекту, на который указывает p. К тому же, указатель \*p не инициализирован.

Задание 4.



Математическая модель:

value1 = 20000

value2 = value1 = 20000

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип |
| value1 | Переменная для хранения значения | long |
| value2 | Переменная для хранения значения | long |
| \*lPtr | Указатель на переменные типа long | long |

Код:

#include <stdio.h>

#include <windows.h>

**int** main(**void**){

**long** value1=20000,value2;

**long** \*lPtr;

lPtr=&value1;

printf("%d**\n**",\*lPtr);

value2=\*lPtr;

printf("%d**\n**",value2);

printf("%p**\n**",&value1);

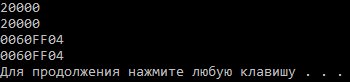
printf("%p**\n**",lPtr);

system("pause");

**return** 0;

}

Результат:



Задание 5.





Т.к. без инициализации указателя бессмысленно применять операцию обращения к объекту по его адресу, единственный вариант исправления таков:

int \*number;

printf("%p\n",number);



В данном случае присвоение невозможно, т.к. указатели ссылаются на переменные разных типов.



Здесь указателю присваивается неинициализированная переменная. Скорее всего такой фрагмент в программе выполнится, но результат будет непредсказуем. Если подразумевалось создание нулевого указателя, то фрагмент будет выглядеть так:

int \*x,y;

x=NULL;



К указателю типа void не применима разадресация. Здесь можно обратиться непосредственно к numPtr:

short \*numPtr,result;

void \*genericPtr=numPtr;

result=\*numPtr+7;



По всей видимости пропущен астериск при выводе.

Возможный вариант:

float x=19.34;

float xPtr=&x;

printf("%f\n",\*xPtr);



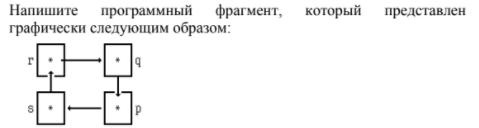
Судя по всему предполагается либо вывод значения указателя - в этом случае ошибка кроется в использовании ненадлежащего спецификатора формата, - либо это следует рассматривать как попытку разыменования указателя с пропущенным астериском – в таком случае ошибка заключается в том, что разыменование не инициализированного указателя невозможно.

Единственное возможное исправление – замена спецификатора формата вывода:

char \*s;

printf("%p\n",s);

Задание 6.



Код:

#include <stdio.h>

#include <locale.h>

**int** main(**void**){

setlocale(LC\_ALL,"Russian");

**int** \*r=NULL,\*s=&r,\*p=&s,\*q=&p;

r=&q;

*/\* test \*/*

printf("Проверка: %d, %d, %d, %d**\n**",r==&q,q==&p,p==&s,s==&r);

getch();

**return** 0;

}

Результат:

