

Самостоятельная работа №7.

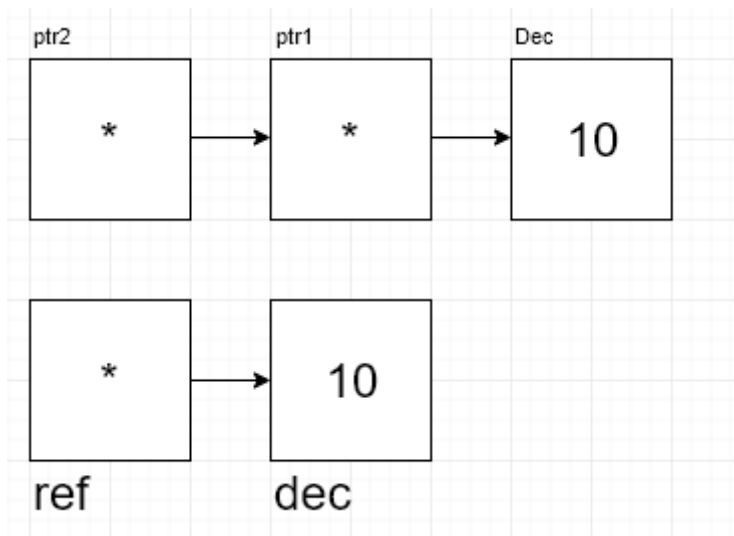
«Указатели на простейшие типы. Операции над указателями»

Задание 1

Постановка задачи:

Каждой строке кода сопоставьте графическое представление.

Результат выполнения:



Задание 2

Постановка задачи:

Дано следующее описание переменных:

```
int *p,*q;  
char *r;
```

Какие из следующих операторов присваивания неправильны и почему?

- (a) `p=q;` (в) `p=NULL;` (д) `q=*p;`
(б) `q=r;` (r) `r=NULL;` (е) `*p=NULL.`

Ответ: б) Тк они ссылаются на объекты разных типов, е) Тк NULL присваивается не указателю а значению хранящемуся по адресу в этом указателе

Задание 3

Постановка задачи:

Дано следующее описание переменных:

```
int *p,*q;  
char *r;
```

Какие из следующих операторов неправильны и почему?

- | | |
|----------------------------------|---------------------------------|
| (a) *r=*p; | (б) *q=(int)*r; |
| (в) if (r!=NULL)
*r=NULL; | (г) if (q>NULL)
*q=*p; |
| (д) if (q==p)
printf("%d",q); | (е) if (q!=r)
scanf("%c",r); |

Ответ: в) Null присваивается значению r, б) Тк присутствует фрагмент (int)

Задание 4

Постановка задачи:

Выполните каждое из следующих заданий, используя для этой цели только один оператор. Считайте объявленными переменные value1 и value2 типа long, причём переменной value1 присвоено значение 20000.

- (а) Объявите указатель IPtr на объект данных типа long.
- (б) Присвойте значение адреса переменной value1 указателю IPtr.
- (в) Выведите значение объекта, на который ссылается IPtr.
- (г) Присвойте значение объекта, на который ссылается IPtr, переменной value2.
- (д) Выведите значение value2.
- (е) Выведите адрес value1.
- (ж) Выведите значение адреса, находящееся в IPtr.

Мат.модель:

Список идентификаторов:

Переменная	Тип	Функция
value1	long	Хранение значения
value2	long	Хранение значения
Ptr1	long(указатель)	Хранение адреса переменной value1

Код:

```

#include<stdio.h>
#include <locale.h>
void main()
{
char*locale = setlocale(LC_ALL, "");
long value2,value1 = 20000;
long *Ptr1; // Пункт а
Ptr1 = &value1; // Пункт б
printf("По адресу: %p хранится значение равное : %ld\n", Ptr1, *Ptr1); // Пункт в
value2 = *Ptr1; // Пункт г
printf("Значение value2 = %ld\n", value2); // Пункт д
printf("Адрес value1: %p \n", &value1); // Пункт е
printf("Адрес в Ptr1: %p ", Ptr1); // Пункт ж
}

```

Результат выполнения:

```

По адресу: 000000000062FE38 хранится значение равное : 20000
Значение value2 = 20000
Адрес value1: 000000000062FE38
Адрес в Ptr1: 000000000062FE38

```

Задание 5

Постановка задачи:

Найдите ошибку в каждом из следующих программных фрагментов. Если ошибку можно исправить, пропишите, как это сделать.

Ответ:

а) Нет ошибки

б) Ошибка в третьей строке, присвоение указателя одного типа другому

в) Попытка присвоить указателю значение переменной. Рабочий вариант:

```
int *x,*y;
```

```
x = y;
```

д) float x=19.34;

```
float *xPtr=&x;
```

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

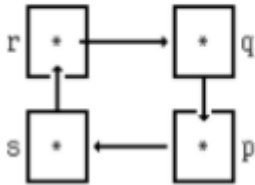
е) char *s;

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

Задание 6

Постановка задачи:

Напишите программный фрагмент, который представлен графически следующим образом:



Код:

```
int *p,*s,*q,*r;
```

```
p = &s;
```

```
s = &r;
```

```
r = &q;
```

```
q = &p;
```