

Лабораторная работа №7.

«Указатели».

Задание № 1.

Постановка задачи:

Заполните пустые места в программе таким образом, чтобы программа выдавала полученный результат.

Код:

```
#include<stdio.h>
#include<conio.h>
#include <locale.h>
void main()
{
    char*locale = setlocale(LC_ALL, "");
    int a,b,c,max;
    int *pa = &a, /* Переменная pa содержит адрес переменной a */
    *pb = &b, /* Переменная pb содержит адрес переменной b */
    *pc = &c, /* Переменная pc содержит адрес переменной c */
    *min;
    printf("Введите значения a,b,c: ");
    scanf("%d %d %d",&a,&b,&c);
    printf("Посмотрим, по каким адресам лежат эти значения:\n");
    printf(" в ячейке с адресом %p лежит %d\n",pa,a);
    printf(" в ячейке с адресом %p лежит %d\n",pb,b);
    printf(" в ячейке с адресом %p лежит %d\n",pc,c);
    printf("Найдем большее из чисел a, b, c и положим\n");
    printf("в ячейку с меньшим адресом.\n");
    /* Поиск наименьшего адреса */
    if ((pa<=pb) && (pa <= pc))
        min=pa;
    else
    {
        if (pb <= pc)
            min = pb;
        else min = pc;
    }
    if (*pa<=*pb)
        max=*pb; // перед max не стоит звездочка, тк мы присваиваем значение указателя переменной
    else max=*pa;
    if (max<=*pc)
        max=*pc;
    /* Требуемое размещение */
    *min = max;
    printf("Значение max=%d расположено по адресу %p.\n\n", max,min);
    getch();
}
```

Вывод:

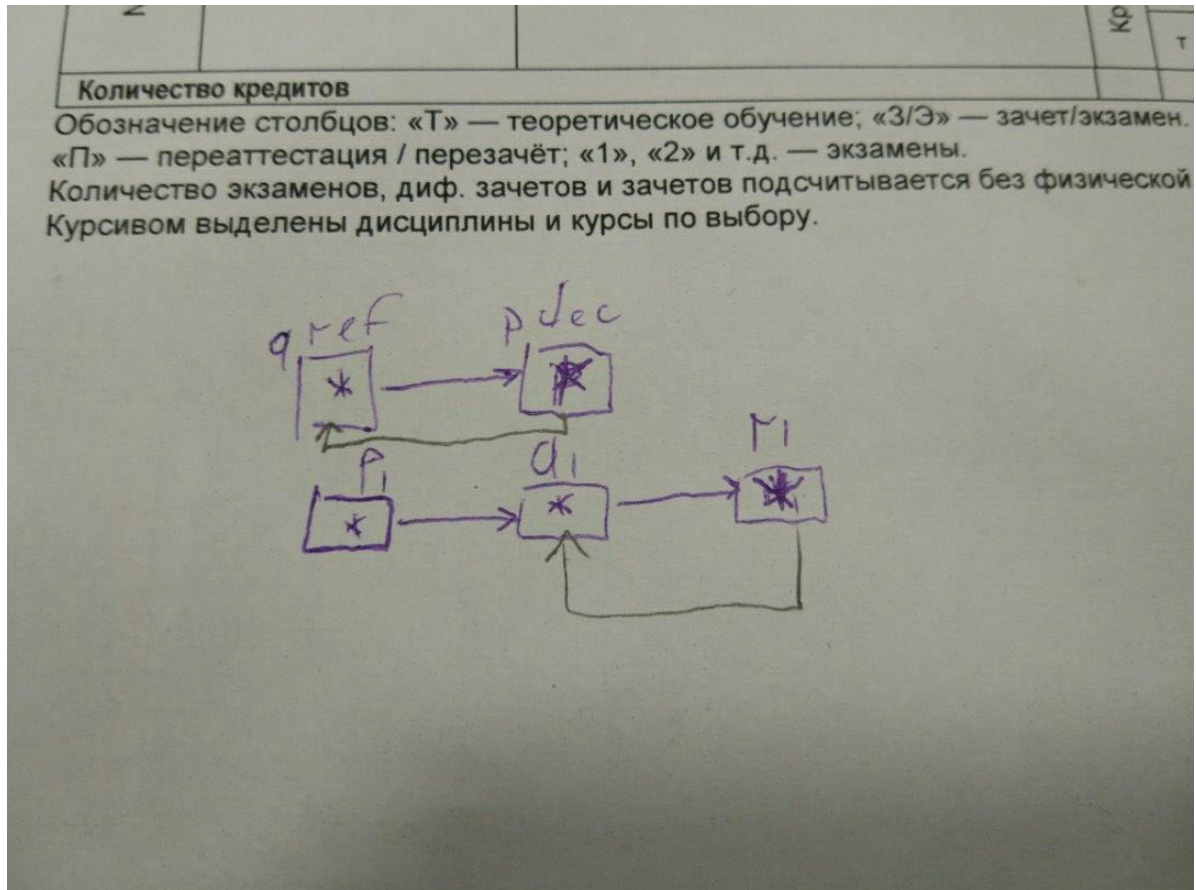
```
Введите значения a,b,c: 3
4
5
Посмотрим, по каким адресам лежат эти значения:
 в ячейке с адресом 000000000062FE1C лежит 3
 в ячейке с адресом 000000000062FE18 лежит 4
 в ячейке с адресом 000000000062FE14 лежит 5
Найдем большее из чисел a, b, c и положим
в ячейку с меньшим адресом.
Значение max=5 расположено по адресу 000000000062FE14.
```

Задание № 2.

Постановка задачи:

Графически представьте взаимосвязь указателей и их значений.

Вывод:



Задание № 3.

Постановка задачи:

Найдите ошибку в использовании функции sizeof.

Код:

```
#include<stdio.h>
#include<conio.h>
#include <locale.h>
void main()
{
    char*locale = setlocale(LC_ALL, "");
    char c;
    int i;
    /* _____ */
    printf("Размер типа char = %2d\n", sizeof(char));
    printf("Размер типа int = %2d\n", sizeof(int));
    printf("Размер переменной типа char =%2d\n", sizeof(c));
    printf("Размер переменной типа int =%2d\n", sizeof(i));
    getch();
}
```

Вывод:

```
Размер типа char = 1
Размер типа int = 4
Размер переменной типа char = 1
Размер переменной типа int = 4
```

Задание № 4.

Постановка задачи:

Ознакомьтесь с программой, заполните пустые места в программе.
Запустите код и сделайте скриншот результата.
Ответьте на комментарии к программе.

Код:

```
/* Демонстрация разыменования указателя, который не */
/* был инициализирован. Это может привести: */
/* (1) к фатальной ошибке во время выполнения прог- */
/* раммы; */
/* (2) случайному искажению данных, в результате */
/* чего программа завершается с неверным результатом */
/* ----- */
#include<stdio.h>
#include<conio.h>
#include <locale.h>
void main()
{
    char*locale = setlocale(LC_ALL, "");
    int dec=10,
    *ptr=&dec, /* указатель ptr инициализирован */
    *ptr1; /* указатель ptr1 не инициализирован */
    /* ----- */
    printf("Указатель инициализирован:\n");
    printf(" По адресу (%p) лежит переменная = %d\n",ptr,dec);
    printf(" &*ptr=%p\n *&ptr=%p\n",&ptr,*&ptr); /*что происходит при
использовании следующих операций *&ptr,*&ptr
*/
    printf("Указатель не инициализирован:\n");
    printf("По адресу (%p) лежит переменная = %d\n",ptr1);
    printf(" &*ptr1=%p\n *&ptr1=%p\n",&ptr1,*&ptr1);
    /* ----- */
    printf("Инициализируем неинициализированный указатель:\n");
    ptr1=&dec; // указателю присваивается адрес переменной dec
    printf("(%p)=%d\n",ptr1,dec);
    getch();
}
```

Вывод:

```
Указатель инициализирован:
По адресу (00000000000062FE44) лежит переменная = 10
*&ptr=00000000000062FE44
*&ptr=00000000000062FE44

Указатель не инициализирован:
По адресу (0000000000000029) лежит переменная = -1335211744
*&ptr1=0000000000000029
*&ptr1=0000000000000029

Инициализируем неинициализированный указатель:
(00000000000062FE44)=10
```