

## Лабораторная работа 9.

### Операции с указателями.

1.

Постановка задачи: Объявите две переменные целого типа. На каждую из них ссылается указатель. На первую ссылается указатель p\_1, а на вторую указатель p\_2. Кроме того, объявлена переменная типа double и указатель на неё p\_dbl.

Используя указатели, подсчитать частное целых переменных (первую делим на вторую) и сохранить это значение в переменную, на которую ссылается p\_dbl.

Ответ:

```
#include<stdio.h>

void main()
{
    int a=2,b=1, *p_1=&a,*p_2=&b;

    double x, *p_dbl=&x;

    *p_dbl=*p_1/(*p_2);

    printf("%f",*p_dbl);
}
```

2.

Постановка задачи: Что выполняет приведенная программа? Найдите ошибку в программе и объясните ее причину. Исправьте программу так, чтобы она работала корректно.

```
#include <stdio.h>

void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}

int main(){
    int x = 3,y = 5;

    printf("x=%d \t y=%d \n",x,y);
```

```

swap(x,y);

printf("x=%d \t y=%d \n",x,y);

return (0);

}

```

Ответ: Данная программа должна была поменять местами 2 переменные, не работает т.к. переменные меняются местами внутри функции, вот рабочий код:

```

#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main(){
    int x = 3,y = 5;
    printf("x=%d \t y=%d \n",x,y);
    swap(&x,&y);
    printf("x=%d \t y=%d \n",x,y);
    return (0);
}

```

3.

Постановка задачи: Что выполняется в данной программе? Дайте ответ и напишите комментарии к программе.

Ответ: #include <stdio.h>

```

void main() {
    int length;
    char *p1, *p2;
    char tmp;
    float a = 5.0f;
    float b = 3.0f;

    printf("a = %.3f\n", a);//вывод с 3 знаками после запятой
    printf("b = %.3f\n", b);//вывод с 3 знаками после запятой
    p1 = (char*) &a;//приведение адреса а к типу char

```

```

p2 = (char*) &b;//приведение адреса b к типу char
length = sizeof(float);//получение максимального значения типа float
while (length--) {//цикл кол-во итераций = длина типа float
tmp = *p1;
*p1 = *p2;
*p2 = tmp;//значение адресов указателей меняются местами
p1++;//перемещение указателя на следующую переменную
p2++;//перемещение указателя на следующую переменную
}
printf("a = %.3f\n", a);
printf("b = %.3f\n", b);
//getch();
}
4.

```

Постановка задачи: Допустимо ли в Си? Если "да" - опишите семантику каждого правильного действия (не принимая во внимание ошибочные); если "нет" - объясните почему.

Ответ:

a). #include <stdio.h>

```
void main() {
```

```
    int i, *p, j, *q;
```

```
    p = &i; q = &p; //получение указателями значений адресов переменных i и p
```

```
    j = *p = 1;//присваивание переменной i=1 и j=i=1
```

```
    q = p-1;//присваивание указателю q первый свободный адрес перед i
```

```
    *p += 1;//присваивание i значения следующей переменной после i
```

```
    i = *++q + *p;//присваивание i значения следующей переменной за i + сама i
```

```
    q -= 1;//присваивание q значения адреса переменной перед p
```

```
    i = *q++ + *q;//присваивание i значения пред. Адрес i + пред. Адрес i ( после происходит
смещение значения q на след адрес = адрес i)
```

```
    printf("i=%d, j=%d, *p=%d, *q=%d \n", i, j, *p, *q);
```

```
}
```

b). int x = 1, y;

```
char c = 'a'; int *pi, *qi; char *pc;
```

```
pi = &x; *pi = 3; y = *pi; *pi = c; qi = pi;
pc = qi; *qi+=1; pi++; * (-- pi) = 5; y = *qi+1;
pc = &c; ++*pc; (*pc)++; *pc++; *pc+=1;
x = (int)pi; pi=(int*)pc; pi=(int*)x; x = 1+ *pi; pc=(char*)pi;
c = *pc; pc = &y; x = qi - pi; qi = 0; qi+=pi;
y = &pi; y = (int)&pi; pi = pi +5; *(pi+1)=0; pi=&(x+0);
```

//в данном примере происходит попытка работы с переменными одного типа но разного размера (судя по моему компилятору)

```
main.c:8:7: warning: cast from pointer to integer of different size
[-Wpointer-to-int-cast] x = (int)pi; pi=(int*)pc; pi=(int*)x; x = 1+ *pi;
pc=(char*)pi; ^
```