

Лабораторная работа 10.

Рекурсия.

1.

Постановка задачи: Дан “нерекурсивный” код (итерационный вариант). Измените код таким образом, чтобы он включал в себя рекурсию функции (избавьтесь от явной итерации).

Ответ:

```
#include <stdio.h>

void call()
{
    char ch;

    printf("Введите символ (Q - окончание): ");
    scanf("%c",&ch);
    printf("Адрес ch=%c равен %u.\n",ch,&ch);
    if (ch != 'Q')
        call();
}

void main()
{
    printf("mda");
    call();
}
```

2.

Постановка задачи: Объясните почему: если использовать нижеописанный код в форме итерации, то мы получаем одинаковый адрес переменной при каждой итерации, а в случае рекурсии – адрес переменной всегда будет изменяться.

```
printf("Введите символ (Q - окончание): ");
fflush(stdin); /* Очистка содержимого выходного буфера */
printf("Адрес ch=%c равен %u.\n",ch=getchar(),&ch);
```

Ответ:

По моим наблюдениям если использовать вышеописанный код хоть при итерациях, хоть при рекурсии не будет получаться выходных данных из задания, но можно предположить, что дело в стэке данных, при цикле происходит переприсваивание на тот же адрес, а во время рекурсии адрес памяти занят и используется в стэке, поэтому следующая переменная присваивается на следующий адрес.

3.

Постановка задачи: Дан “нерекурсивный” код (итерационный вариант). Измените код таким образом, чтобы он включал в себя рекурсию пользовательской функции (избавьтесь от явной итерации).

Ответ: #include<stdio.h>

```
void pcall (int a)
{
    if (a != 0)
        pcall(a-1);
    printf("%5u, ",a);
}
```

```
void main()
{
    unsigned int n;
    printf("Введите натуральное число: "); scanf("%u",&n);
    pcall(n-1);
}
```

4.

Постановка задачи: Дан “нерекурсивный” код (итерационный вариант). Измените код таким образом, чтобы он включал в себя рекурсию пользовательской функции (избавьтесь от явной итерации).

Ответить на вопрос: «Какое математическое вычисление выполняет код»

Ответ:

#include<stdio.h>

```
long fact (int a)
{
    if (a != 0)
        return a*fact(a-1);
    else
        return 1;
}

void fprint(int a)
{
    unsigned f;
    if (a!=0)
        fprint(a-1);
    f=fact(a);
    printf("%2ld!=%ld\n",a,f);
}

void main()
{
    long i;
    fprint(10);
}
```

Это подсчет факториала