

[Home](#) | [Profile](#) | [My Courses](#) | [All Courses](#) | [Feedback!](#)

Welcome niloy@cse.iitkgp.ernet.in | [Settings](#) | [Logout](#)



Assignment saved below.

- [All Assignments](#)
- [Assignment 2/09/2008](#)
- [Submissions](#)

Assignment 2/09/2008

**Deadline:** 02 Sep 00:00

### Assignment 1

Write a C code to compute the reverse of an integer using a recursive function. The function will return to the main() the value of the reverse and the main() shall print the value of the reverse. The main() should scanf() the integer and call the function. Identify the base case when you need to terminate the recursive call.

Name the function as given in the code skeleton below.

The code skeleton should look something like this:

```
int main()
{
    .
    .
    scanf("%d",&n);
    rev=reverse(n);
    printf("%d\n",rev);
}

int reverse(int n)
{
    .
    .
    .
}
```

## Assignment 2

Write a C program that takes in as input 2 positive numbers ( The input in the testcases will be positive hence no need to check for negative numbers) and compute their GCD (Greatest Common Divisor also called HCF- Highest Common Factor).The GCD is computed recursively as:

$$\begin{aligned} \text{gcd}(x,y) &= \text{gcd} ( x \% y , y) && \text{if } x > y \\ &= \text{gcd} ( y \% x , x) && \text{if } y > x \\ &= y && \text{if } x = 0 \end{aligned}$$

The code skeleton should look something like this:

```
int main()
{
    .
    .
    scanf("%d",&n);
    scanf("%d",&m);
    euclidean_gcd = gcd ( n , m);
    printf("%d\n",euclidean_gcd);
}

int gcd (int x, int y)
{
    .
    .
    .
}
```

## Assignment 3

Write a C program to compute  ${}^n\text{Cr}$  using recursions. Recall that  ${}^n\text{Cr}$  is  $(n!)/((n-r)! * r!)$  . Also  ${}^n\text{Cr}$  can be written in terms of  ${}^{n-1}\text{Cr}-1$  as :

$${}^n\text{Cr} = [ n * {}^{n-1}\text{Cr}_{-1} ] / r$$

${}^n\text{Cr}$  is defined only for  $n \geq r$ . Assume that in the testcases  $n$  is always greater than or equal to  $r$ .

Identify the base cases when you need to terminate the recursive calls.

The code skeleton should look like this:

```
int main()  
{  
    .  
    .  
    .  
    scanf ( "%d",&n);  
    scanf ( "%d",&r);  
    nCr = combinatorics ( n, r );  
    printf ("%d\n",nCr);  
}
```

```
int combinatorics (int n , int r)  
{  
    .  
    .  
    .  
}
```

## Resources

No Attachment has been posted yet.

- [Add File](#)
- [Add HTML](#)
- [Add Link](#)

Here in HTML Panel

[Edit Assignment](#)