**1. Program to Print Prime Numbers between 2 integers.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>

int checkPrimeNumber(int n);

int main() {

int n1, n2, i, flag;

printf("Enter two positive integers: ");

scanf("%d %d", &n1, &n2);

// swap n1 and n2 if n1 > n2

if (n1 > n2) {

n1 = n1 + n2;

n2 = n1 - n2;

n1 = n1 - n2;

}

prinf("Prime numbers between %d and %d are: ", n1, n2);

for (i = n1 + 1; i < n2; ++i) {

// flag will be equal to 1 if i is prime

flag = checkPrimeNumber(i);

if (flag == 1) {

printf("%d ", i);

}

}

return 0;

}

// user-defined function to check prime number

int checkPrimeNumber(int n) {

int j, flag = 1;

for (j = 2; j <= n / 2; ++j) {

if (n % j == 0) {

flag = 0;

break;

}

}

return flag;

}

Output

Enter two positive integers: 12

30

Prime numbers between 12 and 30 are: 13 17 19 23 29

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Find the sum of natural numbers using Recursion**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>

int addNumbers(int n);

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

printf("Sum = %d", addNumbers(num));

return 0;

}

int addNumbers(int n) {

if (n != 0)

return n + addNumbers(n - 1);

else

return n;

}

Output

Enter a positive integer: 20

Sum = 210

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Factorial using Recursion**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<stdio.h>

long int multiplyNumbers(int n);

int main() {

int n;

printf("Enter a positive integer: ");

scanf("%d",&n);

printf("Factorial of %d = %ld", n, multiplyNumbers(n));

return 0;

}

long int multiplyNumbers(int n) {

if (n>=1)

return n\*multiplyNumbers(n-1);

else

return 1;

}

Output

Enter a positive integer: 6

Factorial of 6 = 720

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. GCD using Recursion**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>

int hcf(int n1, int n2);

int main() {

int n1, n2;

printf("Enter two positive integers: ");

scanf("%d %d", &n1, &n2);

printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));

return 0;

}

int hcf(int n1, int n2) {

if (n2 != 0)

return hcf(n2, n1 % n2);

else

return n1;

}

Output

Enter two positive integers: 366

60

G.C.D of 366 and 60 is 6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Power of a number using Recursion.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>

int power(int n1, int n2);

int main() {

int base, a, result;

printf("Enter base number: ");

scanf("%d", &base);

printf("Enter power number(positive integer): ");

scanf("%d", &a);

result = power(base, a);

printf("%d^%d = %d", base, a, result);

return 0;

}

int power(int base, int a) {

if (a != 0)

return (base \* power(base, a - 1));

else

return 1;

}

Output

Enter base number: 3

Enter power number(positive integer): 4

3^4 = 81

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Find nCr using user defined functions**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>

int factorial(int n) {

int factorial = 1;

for (int i = 2; i <= n; i++)

factorial = factorial \* i;

return factorial;

}

int nCr(int n, int r) {

return factorial(n) / (factorial(r) \* factorial(n - r));

}

int main() {

int n = 5, r = 2;

printf("%d", nCr(n, r));

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7. Find nCr using Recursive Functions**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<stdio.h>

#include<conio.h>

int rec\_ncr(int ,int );

void main()

{

int n,r;

printf("Enter n and r");

scanf("%d%d",&n,&r);

printf("The value of %dC%d is %d",n,r,rec\_ncr(n,r));

}

int rec\_ncr(int n,int r)

{

if(r==0|r==n)

return 1;

else

return rec\_ncr(n-1,r-1)+rec\_ncr(n-1,r);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Write a program to initialize an automatic and static variable and increment it in the function. Call this function thrice and print the value of the variable every time after incrementing.

#include<stdio.h>

int fun()

{

static int count = 0;

count++;

return count;

}

int main()

{

printf("%d ", fun());

printf("%d ", fun());

return 0;

}