

1) WAP to display  $n^{\text{th}}$  element of fibonacci series using recursive function.

```
#include <stdio.h>
```

```
int find_fibonacci (int n){
```

```
    if (n == 0){  
        return 0;
```

```
    }
```

```
    else if (n == 1){
```

```
        return 1;
```

```
    }
```

```
    else {
```

```
        return find_fibonacci(n-1) + find_fibonacci(n-2);
```

```
    }
```

```
int main () { int n;
```

```
    printf ("Enter the  $n^{\text{th}}$  number: ");
```

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

```
    int num = find_fibonacci (n);
```

```
    printf ("The nth  $n^{\text{th}}$  term of fibonacci series: %d", n, num);
```

```
    return
```

// The indexing of fibonacci series starts with 0 so first  
th. element is given when  $n = 0$ .

```
    return 0;
```

```
}
```

Output:

Enter the  $n^{\text{th}}$  number: 5

The 5th term of fibonacci series :- 5



#2

Write a function to check whether the number is prime or not and use that function to print prime numbers from 1 to n entered by user.

Date: .....

Page: .....

#include &lt;stdio.h&gt;

void print\_prime(int num, int upper){

int count = 0;

for (int i = 1; i &lt;= num; i++){

if (num % i == 0){

count++;

}

}

if (count == 2){

printf("%d is prime.", num);

}

else{

printf("%d is not prime.", num);

}

printf("\n\n");

int count2 = 0;

for (int i = 1; i &lt;= upper; i++){

for (int j = i; j &gt;= i; j--){

if (i % j == 0){

count2++;

}

}

if (count2 == 2){

printf("%d \t", i);

}

count2 = 0;

}

}



```
int main () {  
    int number, upper-limit;  
  
    printf ("Enter a number to check if prime: ");  
    scanf ("%d", & number);  
  
    printf ("\n Enter a upper limit to where you want prime number: ");  
    scanf ("%d", & upper-limit);  
    printf ("\n\n");  
    print_prime (number, upper-limit);  
  
    return 0;  
}
```

### OUTPUT

Enter a number to check if prime : 7  
Enter a upper limit to where you want prime number: 10

7 is prime

2 3 5 7

#3 Write a program using recursion to compute the sum of the following series:

$$1^2 - 2^2 + 3^2 - 4^2 \dots + (-1)^{n+1} n^2$$

without using pow() function. You should read the value of n from user.

Date \_\_\_\_\_

Page \_\_\_\_\_

```
#include <stdio.h>
```

```
int power (int num, int p) {
```

```
    if (p == 0) {
```

```
        return 1;
```

```
    }
```

```
    else {
```

```
        return num * power ( num, p-1);
```

```
    }
```

```
}
```

```
int sum_of_series (int n) {
```

```
    if (n == 0) {
```

```
        return 0;
```

```
    }
```

```
    else {
```

```
        return power (-1, n+1) * power (n, 2) + sum_of_series (n-1);
```

```
    }
```

```
}
```

```
int main () {
```

```
    int n;
```

```
    printf ("Enter the number of terms: ");
```

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

```
    int sum = sum_of_series (n);
```

```
    printf ("%d", sum);
```

```
    return 0;
```

```
}
```

OUTPUT

Enter the number of terms : 3

6



4

```
#include <stdio.h>
```

```
#include <math.h>
```

```
float frequency (float r, float L, float C) {
    return sqrt((1/(L*C)) - ((r*r)/(4*L*C)));
}
```

```
int main () {
```

```
    float L, r;
```

```
    printf ("Enter the value of inductance and resistance: ");
```

```
    scanf ("%f %f", &L, &r);
```

```
    printf ("Inductance %f Resistance %f Capacitance %f Frequency %f\n", L, r, C, frequency(L, r, C));
```

```
    for (float i = 0.01; i <= 0.1; i += 0.01) {
```

```
        float C = i;
```

```
        float freq = frequency (r, L, C);
```

```
        printf ("%f %f %f %f\n", L, r, C, freq);
    }
```

```
    return 0;
```

```
}
```



## OUTPUT

Enter the value of inductance and resistance: 0.05

0.2

Inductance	Resistance	Capacitance	Frequency
0.050	0.200	0.010	43.589
0.050	0.200	0.020	81.225
0.050	0.200	0.030	25.604
0.050	0.200	0.040	22.220
0.050	0.200	0.050	19.900
0.050	0.200	0.060	18.181
0.050	0.200	0.070	16.849
0.050	0.200	0.080	15.762
0.050	0.200	0.090	14.866
0.050	0.200	0.100	14.107