Q1.Write a program to find the Nth term of the Fibonnaci series.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num, a = 0, b = 1, c;
    printf("Enter number = ");
    scanf("%d", &num);
    for (int i = 2; i < num; ++i)
    {
        c = a + b;
        a = b;
        b = c;
    }
    printf("%d ", c);
    return 0;
}
```

Q2.Write a program to print first N terms of Fibonacci series

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num, a = 0, b = 1, c;
    printf("Enter number = ");
    scanf("%d", &num);
    printf("%d %d ", a, b);
    for (int i = 2; i < num; ++i)
    {
        c = a + b;
        printf("%d ", c);
        a = b;
        b = c;
    }
    return 0;
}
```

Q3.Write a program to check whether a given number is there in the Fibonacci series or not.

```c
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num, a = 0, b = 1, c = 0;
    printf("Enter number = ");
    scanf("%d", &num);
    while (c < num)
    {
        c = a + b;
        a = b;
        b = c;
    }
    if (c == num)
    {
        printf("Number found");
    }
    else
    {
        printf("Number not found");
    }
    return 0;
}
```

Q4.Write a program to calculate HCF of two numbers

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int a, b, c, flag = 0,i;
    printf("Enter two number = ");
    scanf("%d %d", &a, &b);
    if (a > b)
    {
        int tmp = b;
        b = a;
        a = tmp;
    }
    if (b % a == 0)
    {
        printf("HCF = %d", a);
        flag = 1;
    }
    else
        for (i = a / 2; !((a % i==0) && (b % i==0)); i--);
    if (flag == 0)
        printf("HCF = %d", i);
    return 0;
}
```

Q5.Write a program to check whether two given numbers are co-prime numbers or not

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int a, b, c, flag = 0, i;
    printf("Enter two number = ");
    scanf("%d %d", &a, &b);
    if (a > b)
    {
        int tmp = b;
        b = a;
        a = tmp;
    }
    if (b % a == 0)
    {
        printf("number is not co-prime");
        flag = 1;
    }
    else
        for (i = a / 2; !((a % i == 0) && (b % i == 0)); i--);
    if (i == 1)
        printf("number is co-prime");
    else if (i > 1)
```

```
        printf("number is not co-prime");
    return 0;
}
```

Q6.Write a program to print all Prime numbers under 100

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int flag = 0;
    for (int i = 2; i <= 100; i++)
    {
        flag = 0;
        for (int j = 2; j <= i / 2; j++)
            if (i % j == 0)
            {
                flag = 1;
                break;
            }
        if (flag == 0)
            printf("%d ", i);
    }
    return 0;
}
```

Q7.Write a program to print all Prime numbers between two given numbers

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num1, num2, i = 2, flag = 0;
    printf("Enter two numbers = ");
    scanf("%d %d", &num1, &num2);
    while (num1 <= num2)
    {
        while (i <= num1 / 2)
        {
            if (num1 % i != 0)
                i++;
            else
            {
                flag = 1;
                break;
            }
        }
        if (flag == 1)
            flag = 0;
        else if ((num1 != 0) && (num1 != 1))
            printf("%d ", num1);
        num1++;
        i=2;
    }
    return 0;
}
```

Q8.Write a program to find next Prime number of a given number

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num, flag = 0;
    printf("Enter number = ");
    scanf("%d", &num);
    while (flag == 0)
    {
        int i = 2;
        num = num + 1;
        while (i <= num / 2)
        {
            if (num % i == 0)
                break;
            else
                i++;
        }
        if ((i > num / 2) && (num > 1))
        {
            flag = 1;
            printf("%d", num);
        }
    }
    return 0;
}
```

Q9.Write a program to check whether a given number is an Armstrong number or not

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num, rem, cube = 0;
    printf("Enter number = ");
    scanf("%d", &num);
    int temp = num;
    while (num)
    {
        rem = num % 10;
        cube = cube + (rem * rem * rem);
        num = num / 10;
    }
    if ((temp == cube) && (temp != 000))
        printf("Number is armstrong");
    else
        printf("Number is not armstrong");
    return 0;
}
```

Q10. Write a program to print all Armstrong numbers under 1000

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    int num = 100, cube, rem;
    while (num <= 1000)
    {
        int temp = num;
        cube = 0;
        while (temp)
        {
            rem = temp % 10;
            cube = cube + (rem * rem * rem);
            temp = temp / 10;
        }
        if (cube == num)
            printf("%d ", cube);
        num++;
    }
    return 0;
}
```