**Assignment - 7 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**(Solution)**

**Iterative Control Statements (Part - 2)**

**1. Write a program to find the Nth term of the Fibonacci series.**

#include <stdio.h>

*int* main()

{

*int* n, i, t1 = 0, t2 = 1, next = 0;

    printf("Enter the n value: ");

    scanf("%d", &n);

    if (n == 0 || n == 1)

        printf("%d", n);

    else

        next = t1 + t2;

    for (i = 3; i <= n; ++i)

    {

        t1 = t2;

        t2 = next;

        next = t1 + t2;

    }

    printf("%d", t2);

    return 0;

}

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

#include <stdio.h>

*int* main()

{

*int* t1 = 0, t2 = 1, t3, n, count = 0;

    printf("Enter the number to print 1st N terms of Fibonacci series: \n");

    scanf("%d", &n);

    printf("First %d FIBONACCI numbers are ...\n", n);

    printf("%d ", t1);

    printf("%d ", t2);

    count = 2;

    while (count < n)

    {

        t3 = t1 + t2;

        count++;

        printf("%d ", t3);

        t1 = t2;

        t2 = t3;

    }

    return 0;

}

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

#include <stdio.h>

*int* main()

{

*int* i, n, t1 = 0, t2 = 1, next = 0;

    printf("Enter a number : ");

    scanf("%d", &n);

    for (i = 0; 1; i++)

    {

        next = t1 + t2;

        t1 = t2;

        t2 = next;

        if (next == n)

        {

            printf("Yes , %d is in fibinacci series.", n);

        }

        if (next > n)

        {

            printf("%d is not  in fibinacci series.", n);

            break;

        }

    }

    return 0;

}

**4. Write a program to calculate HCF of two numbers**

#include <stdio.h>

*int* main()

{

*int* a, b, i, hcf;

    printf("Enter two numbers: ");

    scanf("%d , %d ", &a, &b);

*int* min = a < b ? a : b;

    for (i = 1; i <= min; i++)

    {

        if ((a % i == 0) && (b % i == 0))

        {

            hcf = i;

        }

    }

    printf("HCF is %d", hcf);

    return 0;

}

**5. Write a program to check whether two given numbers are co-prime**

**numbers or not**

#include <stdio.h>

*int* main()

{

*int* a, b, i, hcf;

    printf("Enter two numbers: ");

    scanf("%d , %d ", &a, &b);

*int* min = a < b ? a : b;

    for (i = 1; i <= min; i++)

    {

        if ((a % i == 0) && (b % i == 0))

        {

            hcf = i;

        }

    }

    printf("HCF is %d", hcf);

    return 0;

}

**6. Write a program to print all Prime numbers under 100**

#include <stdio.h>

*int* main()

{

*int* x, i;

    for (x = 1; x <= 100; x++)

    {

        for (i = 2; i < x; i++)

        {

            if (x % i == 0)

            {

                break;

            }

        }

        if (i == x)

        {

            printf("%d ", x);

        }

    }

    return 0;

}

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

#include <stdio.h>

*int* main()

{

*int* a, b, x, i;

    printf("Enter two numbers : \n");

    scanf("%d %d", &a, &b);

    for (x = a + 1; x <= b - 1; x++)

    {

        for (i = 2; i < x; i++)

        {

            if (x % i == 0)

            {

                break;

            }

        }

        if (i == x)

        {

            printf("%d ", x);

        }

    }

    return 0;

}

**8. Write a program to find next Prime number of a given number**

#include <stdio.h>

*int* main()

{

*int* i, n, x, a = 0;

    printf("Enter the number : \n");

    scanf("%d", &x);

    for (n = x; 1; n++)

    {

        a = 0;

        for (i = 2; i <= n / 2; i++)

        {

            if (n % i == 0)

            {

                a = 1;

            }

        }

        if (a == 0)

        {

            printf("%d ", n);

            break;

        }

    }

    return 0;

}

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

#include <stdio.h>

*int* main()

{

*int* num, temp, remainder, result = 0;

    printf("Enter a three-digit integer: ");

    scanf("%d", &num);

    temp = num;

    while (temp != 0)

    {

        // remainder contains the last digit

        remainder = temp % 10;

        result += remainder \* remainder \* remainder;

        // removing last digit from the original number

        temp /= 10;

    }

    if (result == num)

        printf("%d is an Armstrong number.", num);

    else

        printf("%d is not an Armstrong number.", num);

    return 0;

}

**10. Write a program to print all Armstrong numbers under 1000**

#include <math.h>

#include <stdio.h>

*int* main()

{

*int* i, sum, num, count = 0;

    printf(

        "All Armstrong number between 1 and 1000 are:\n");

    // This loop will run for 1 to 1000

    for (i = 1; i <= 1000; i++)

    {

        num = i;

        // Count number of digits.

        while (num != 0)

        {

            num /= 10;

            count++;

        }

        num = i;

        sum = pow(num % 10, count) + pow((num % 100 - num % 10) / 10, count) + pow((num % 1000 - num % 100) / 100, count);

        // Check for Armstrong Number

        if (sum == i)

        {

            printf("%d ", i);

        }

        count = 0;

    }

}