**Exp 1:** WAP for printing Fibonacci sequence. Take input from the user to print up to a certain limit.

**Solution:-**

**Code:-**

#include <stdio.h>

/\*Fabinaaci sequence\*/

void printFibonacci(int limit);

int main()

{

int limit;

printf("Enter the number of Fibonacci terms to print: ");

scanf("%d", &limit);

if (limit < 1)

{

printf("Please enter a positive number.\n");

} else

{

printFibonacci(limit);

}

return 0;

}

void printFibonacci(int limit)

{

int a = 0, b = 1, c;

printf("Fibonacci Sequence up to %d terms:\n", limit);

/\*Special case for the first two terms\*/

if (limit >= 1)

{

printf("%d ", a);

}

if (limit >= 2)

{

printf("%d ", b);

}

/\*Generate and print the remaining terms\*/

for (int i = 3; i <= limit; i++)

{

c = a + b;

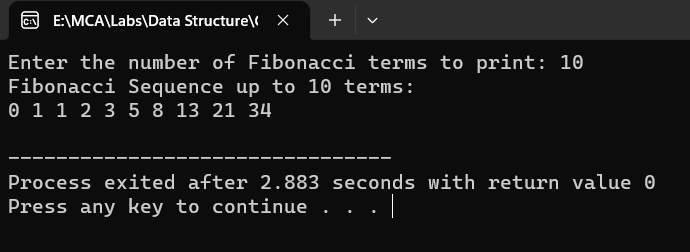
printf("%d ", c);

a = b;

b = c;

}

printf("\n");

****}

**Output:-**

**Exp 2:** WAP to swap two variables without using a third variable, depict the same using call by value concept.

**Solution:-**

**Code:-**

#include <stdio.h>

/\* Function to swap two variables using call by value\*/

void swap(int a, int b);

int main()

{

int x, y;

printf("Enter two integers: ");

scanf("%d %d", &x, &y);

/\*Call the swap function\*/

swap(x, y);

return 0;

}

void swap(int a, int b)

{

printf("Before swapping: a = %d, b = %d\n", a, b);

a = a + b;

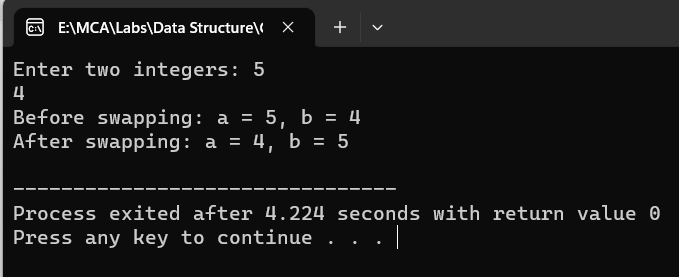
b = a - b;

a = a - b;

printf("After swapping: a = %d, b = %d\n", a, b);

}

**Output:-**

****

**Exp 3:** A positive integer is entered through the keyboard. Write a Function to print the prime factors of this number.

For example, 24 have prime factors: 2,2,2, and 3, whereas 35 have prime factors 5 and 7.

**Solution:-**

**Code:-**

#include <stdio.h>

/\*Function to print prime factors of a number\*/

void primeFactors(int n);

int main()

{

    int num;

    printf("Enter a positive integer: ");

    scanf("%d", &num);

    if (num <= 0)

     {

    printf("Please enter a positive integer.\n");

    } else

     {

        printf("Prime factors of %d are: ", num);

        primeFactors(num);

        printf("\n");

    }

    return 0;

}

// Function to print prime factors of a number

void primeFactors(int n)

 {

    // Print the number of 2s that divide n

    while (n % 2 == 0)

     {

        printf("2 ");

        n = n / 2;

    }

    // n must be odd at this point, so a skip of 2 (i = i + 2) can be used

    for (int i = 3; i \* i <= n; i = i + 2)

     {

        // While i divides n, print i and divide n

        while (n % i == 0)

         {

            printf("%d ", i);

            n = n / i;

        }

    }

    // If n is still greater than 1, then it is a prime number itself

    if (n > 1)

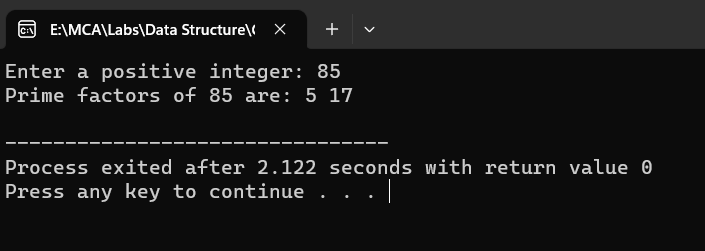
     {

        printf("%d ", n);

    }

}

**Output:-**

****

**Ex: 4** WAP which makes use of Switch and functions to implement the conversion of a given number to a given format.

**Solution:-**

**Code:-**

#include <stdio.h>

/\* Function to convert decimal to binary\*/

void decimalToBinary(int decimal);

int main()

{

    int decimal;

    printf("Enter a decimal number: ");

    scanf("%d", &decimal);

    if (decimal < 0) {

        printf("Please enter a non-negative decimal number.\n");

    } else {

        decimalToBinary(decimal);

    }

    return 0;

}

/\* Function to convert decimal to binary\*/

void decimalToBinary(int decimal) {

    int binary[32]; // Assuming a 32-bit binary representation

    int index = 0;

    /\* Handle the case of 0 explicitly\*/

    if (decimal == 0) {

        printf("Binary: 0\n");

        return;

    }

    /\* Convert decimal to binary\*/

    while (decimal > 0) {

        binary[index++] = decimal % 2;

        decimal = decimal / 2;

    }

    /\* Print the binary representation in reverse order\*/

    printf("Binary: ");

    for (int i = index - 1; i >= 0; i--) {

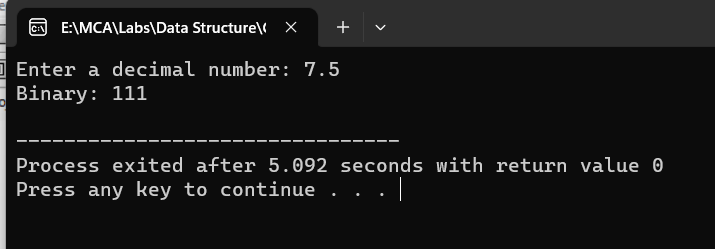
        printf("%d", binary[i]);

    }

    printf("\n");

}

**Output:-**

****