Fibonacci Series in C using loop

A simple for loop to display the series. Program prompts user for the number of terms and displays the series having the same number of terms.

```
#include<stdio.h>
int main()
{
    int count, first_term = 0, second_term = 1, next_term, i;

    //Ask user to input number of terms
    printf("Enter the number of terms:\n");
    scanf("%d",&count);

    printf("First %d terms of Fibonacci series:\n",count);
    for ( i = 0 ; i < count ; i++ )
    {
       if ( i <= 1 )
          next_term = i;
       else
       {
          next_term = first_term + second_term;
          first_term = second_term;
          second_term = next_term;
       }
       printf("%d\n",next_term);
    }

    return 0;
}
```

Using this program you can find out the prime numbers between 1 to 100, 100 to 999 etc. You just need to input the range, for e.g. if you want the prime numbers from 100 to 999 then enter numbers 100 and 999 when program prompts for input.

// Program to find prime numbers in Range

```
#include <stdio.h>
int main()
{
   int num1, num2, flag_var, i, j;

   /*  Ask user to input the from/to range
        like 1 to 100, 10 to 1000 etc.  */
   printf("Enter range: ");
   //Store the range in variables using scanf
   scanf("%d %d", &num1, &num2);

   if(num1>num2)
   {
      int temp= num1;
      num1= num2;
      num2= temp;
   }

   //Display prime numbers for input range (Exclusive)
   printf("Prime numbers from %d and %d are:\n", num1, num2);

   for(i=num1; i<=num2; ++i)
   {
```

```
        flag_var=0;
        for(j=2; j<=i/2; ++j)
        {
            if(i%j==0)
            {
                flag_var=1;
                break;
            }
        }

        if(flag_var==0 && i>1)
            printf("%d\n",i);
    }

    return 0;
}
```

C Program to check Armstrong number

A number is called as Armstrong number if sum of cubes of digits of number is equal to the number itself. In the below C program, we are checking whether the input number is Armstrong or not.

```
#include<stdio.h>
int main()
{
   int num,copy_of_num,sum=0,rem;

   //Store input number in variable num
   printf("\nEnter a number:");
   scanf("%d",&num);

   /* Value of variable num would change in the
      below while loop so we are storing it in
      another variable to compare the results
      at the end of program
   */
   copy_of_num = num;

   /* We are adding cubes of every digit
    * and storing the sum in variable sum
    */
   while (num != 0)
   {
      rem = num % 10;
      sum = sum + (rem*rem*rem);
      num = num / 10;
   }

   /* If sum of cubes of every digit is equal to number
    * itself then the number is Armstrong
    */
   if(copy_of_num == sum)
```

```
    printf("\n%d is an Armstrong Number",copy_of_num);
  else
    printf("\n%d is not an Armstrong Number",copy_of_num);
  return(0);
}
```

Output:

Enter a number: 370
370 is an Armstrong Number
You can verify the result like this:
370 = 3*3*3 + 7*7*7 + 0*0*0
= 27 + 343 + 0
= 370

As you can see that sum of digits of number 370 is equal to the number itself.

Program 1: check palindrome using while loop

```
/* Program to check if a number is palindrome or not
 * using while loop
 */
```

```
#include <stdio.h>
int main()
{
   int num, reverse_num=0, remainder,temp;
   printf("Enter an integer: ");
   scanf("%d", &num);

   /* Here we are generating a new number (reverse_num)
    * by reversing the digits of original input number
    */
   temp=num;
   while(temp!=0)
   {
      remainder=temp%10;
      reverse_num=reverse_num*10+remainder;
      temp/=10;
   }

   /* If the original input number (num) is equal to
    * to its reverse (reverse_num) then its palindrome
    * else it is not.
    */
   if(reverse_num==num)
      printf("%d is a palindrome number",num);
   else
      printf("%d is not a palindrome number",num);
   return 0;
}
```

Array

```
/*
Write a C Program
to input ten numbers in array
of ten integers. Display all values from array
*/
#include<stdio.h>

int main()
{
   int numArray[10], i;
   //input in array using for loop
   for(i=0; i<10; i++)
      {
           printf("Enter Element No:%d=",i+1);
       scanf("%d", &numArray[i]);
      }
   // show array elements using for loop
    printf("The elements of array are:\n");
    for(i=0;i<10;i++)
      printf("%d\n", numArray[i]);

   return 0;
}
```

C program to declare, initialize, input and print array elements - Codeforwin

Write a C program to declare, initialize, input elements in array and print array. How to input and display elements in an array using for loop in C programming. C program to input and print array elements using loop.

Example

Input

Input size: 10
Input elements: 1
2
3
4
5
6
7
8
9
10
Output

Output: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

How to input and print array elements?

Array uses an index for accessing an element. Array index starts from 0 to N-1 (where N is the number of elements in array).

Array and array index representation

To access any an array element we use.

array[0] = 10
array[1] = 20
array[2] = 30

array[9] = 100
Since array index is an integer value. Hence, rather hard-coding constant array index, you can use integer variable to represent index. For example,

int i = 0;
array[i] = 10; // Assigns 10 to first array element
Program to input and print array elements

```
/**
 * C program to read and print elements in an array
 */

#include <stdio.h>
#define MAX_SIZE 1000 // Maximum array size

int main()
{
    int arr[MAX_SIZE]; // Declare an array of MAX_SIZE
    int i, N;

    /* Input array size */
    printf("Enter size of array: ");
    scanf("%d", &N);

    /* Input elements in array */
    printf("Enter %d elements in the array : ", N);
    for(i=0; i<N; i++)
    {
        scanf("%d", &arr[i]);
    }
```

```
    /*
     * Print all elements of array
     */
    printf("\nElements in array are: ");
    for(i=0; i<N; i++)
    {
        printf("%d, ", arr[i]);
    }

    return 0;
}
```

Note: Using i < N is equivalent to i <= N-1.

Advance your skills by learning this using recursive approach.

The above method uses array notation to print elements. You can also use pointer notation to access an array in C. The statement arr[i] is equivalent to *(arr + i).

Output

```
Enter size of array: 10
Enter 10 elements in the array : 10
20
30
40
50
60
70
80
90
100
```

Elements in array are : 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
Happy coding 😉

```
/**
 * C program to count total number of even and odd elements in an array
 */

#include <stdio.h>

#define MAX_SIZE 100 //Maximum size of the array

int main()
{
    int arr[MAX_SIZE];
    int i, size, even, odd;

    /* Input size of the array */
    printf("Enter size of the array: ");
    scanf("%d", &size);

    /* Input array elements */
    printf("Enter %d elements in array: ", size);
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    /* Assuming that there are 0 even and odd elements */
```

```
    even = 0;
    odd  = 0;

    for(i=0; i<size; i++)
    {
        /* If the current element of array is even then increment even
count */
        if(arr[i]%2 == 0)
        {
            even++;
        }
        else
        {
            odd++;
        }
    }

    printf("Total even elements: %d\n", even);
    printf("Total odd elements: %d", odd);

    return 0;
}
```

```
   /*
     * C program to accept N numbers and arrange them in an
ascending order
     */
```

```
#include <stdio.h>
void main()
{

    int i, j, a, n, number[30];
    printf("Enter the value of N \n");
    scanf("%d", &n);

    printf("Enter the numbers \n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);

    for (i = 0; i < n; ++i)
    {

        for (j = i + 1; j < n; ++j)
        {

            if (number[i] > number[j])
            {

                a =  number[i];
                number[i] = number[j];
                number[j] = a;

            }

        }

    }
```

```
        printf("The numbers arranged in ascending order are given
below \n");
        for (i = 0; i < n; ++i)
            printf("%d\n", number[i]);

    }



**
/*
 * C program to input N numbers and store them in an array.
 * Do a linear search for a given key and report success
 * or failure.
 */
#include <stdio.h>

void main()
{  int num;

    int i,  keynum, found = 0;

    printf("Enter the number of elements ");
    scanf("%d", &num);
    int array[num];
    printf("Enter the elements one by one \n");
    for (i = 0; i < num; i++)
    {
        scanf("%d", &array[i]);
```

```
    }

    printf("Enter the element to be searched ");
    scanf("%d", &keynum);
    /*  Linear search begins */
    for (i = 0; i < num ; i++)
    {
        if (keynum == array[i] )
        {
            found = 1;
            break;
        }
    }
    if (found == 1)
        printf("Element is present in the array at position %d",i+1);
    else
        printf("Element is not present in the array\n");
}




/*
    * C program to accept a set of numbers and arrange them
    * in a descending order
    */

    #include <stdio.h>
    void main ()
    {
```

```
    int number[30];

    int i, j, a, n;
    printf("Enter the value of N\n");
    scanf("%d", &n);

    printf("Enter the numbers \n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);

    /*  sorting begins ... */

    for (i = 0; i < n; ++i)
    {
       for (j = i + 1; j < n; ++j)
       {
          if (number[i] < number[j])
          {
             a = number[i];
             number[i] = number[j];
             number[j] = a;
          }
       }
    }

    printf("The numbers arranged in descending order are given
below\n");

    for (i = 0; i < n; ++i)
```

```
        {
            printf("%d\n", number[i]);
        }

    }
```

// Large Number

```
#include <stdio.h>

int main()
{

        int size, i, largest;

        printf("\n Enter the size of the array: ");
        scanf("%d", &size);
        int array[size];

        printf("\n Enter %d elements of  the array: \n", size);

        for (i = 0; i < size; i++)
        {
            scanf("%d", &array[i]);
        }

        largest = array[0];
```

```
        for (i = 1; i < size; i++)
        {
            if (largest < array[i])
            largest = array[i];
        }

        printf("\n largest element present in the given array is : %d",
largest);

        return 0;

}


/*
 * C program to read N integers into an array A and
 * a) Find the sum of all numbers
 * b) Find the average of all numbers
 * Display the results with suitable headings
 */

#include  <stdio.h>

int main()
{
    int i, num;
    float total = 0.0, average;
    printf ("Enter the value of N \n");
    scanf("%d", &num);
    int array[num];
```

```
    printf("Enter %d numbers (-ve, +ve and zero) \n", num);

    for (i = 0; i < num; i++)
    {
        scanf("%d", &array[i]);
    }

    printf("Input array elements \n");

    for (i = 0; i < num; i++)
    {
        printf("%+3d\n", array[i]);
    }

    /*  Summation starts */

    for (i = 0; i < num; i++)
    {
        total+=array[i];/* this means total=total+array[i]; */
    }

    average = total / num;

    printf("\n Sum of all numbers =  %.2f\n", total);

    printf("\n Average of all input numbers =  %.2f\n", average);

}
```

A Simple C++ program to add two Matrices
Here we are asking user to input number of rows and columns of matrices and then we ask user to enter the elements of both the matrices, we are storing the input into a multidimensional array for each matrix and after that we are adding corresponding elements of both the matrices and displaying them on screen.

```
#include<iostream>
using namespace std;

int main()
{
   int row, col, m1[10][10], m2[10][10], sum[10][10];

   cout<<"Enter the number of rows(should be >1 and <10): ";
   cin>>row;
   cout<<"Enter the number of column(should be >1 and <10): ";
   cin>>col;
   cout << "Enter the elements of first 1st matrix: ";
   for (int i = 0;i<row;i++ ) {
     for (int j = 0;j < col;j++ ) {
       cin>>m1[i][j];
     }
   }
   cout << "Enter the elements of first 1st matrix: ";
   for (int i = 0;i<row;i++ ) {
     for (int j = 0;j<col;j++ ) {
       cin>>m2[i][j];
```

```
    }
  }

  cout<<"Output: ";
  for (int i = 0;i<row;i++ ) {
    for (int j = 0;j<col;j++ ) {
      sum[i][j]=m1[i][j]+m2[i][j];
      cout<<sum[i][j]<<" ";
    }
  }

  return 0;
}
```