1. Fibonacci Series

Code:

```
#include<stdio.h>
int main()
{
 int n1=0,n2=1,n3,i,number;
 printf("Enter the number of elements:");
 scanf("%d",&number);
 printf("\n%d %d",n1,n2);//printing 0 and 1
 for(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed
 {
  n3=n1+n2;
  printf(" %d",n3);
  n1=n2;
  n2=n3;
 }
 return 0;
}
```

2. Fibonacci using recursion

Code:

```
#include<stdio.h>
void printFibonacci(int n){
    static int n1=0,n2=1,n3;
    if(n>0){
        n3 = n1 + n2;
        n1 = n2;
        n2 = n3;
```

```
        printf("%d ",n3);
        printFibonacci(n-1);
    }
}
int main(){
    int n;
    printf("Enter the number of elements: ");
    scanf("%d",&n);
    printf("Fibonacci Series: ");
    printf("%d %d ",0,1);
    printFibonacci(n-2);//n-2 because 2 numbers are already printed
  return 0;
 }
```

3. Prime Number

Code:

```
#include<stdio.h>
int main(){
int n,i,m=0,flag=0;
printf("Enter the number to check prime:");
scanf("%d",&n);
m=n/2;
for(i=2;i<=m;i++)
{
if(n%i==0)
{
printf("Number is not prime");
flag=1;
break;
}
```

```
15.}
16.if(flag==0)
17.printf("Number is prime");
18.return 0;
19. }
```

4. Palindrome

Code:

```
1.  #include<stdio.h>
2.  int main()
3.  {
4.  int n,r,sum=0,temp;
5.  printf("enter the number=");
6.  scanf("%d",&n);
7.  temp=n;
8.  while(n>0)
9.  {
10.r=n%10;
11.sum=(sum*10)+r;
12.n=n/10;
13.}
14.if(temp==sum)
15.printf("palindrome number ");
16.else
17.printf("not palindrome");
18.return 0;
19.}
```

5. Factorial

Code:

```c
#include<stdio.h>
int main()
{
 int i,fact=1,number;
 printf("Enter a number: ");
  scanf("%d",&number);
    for(i=1;i<=number;i++){
      fact=fact*i;
  }
 printf("Factorial of %d is: %d",number,fact);
return 0;
}
```

6. Factorial Using Recursion

Code:

```c
#include<stdio.h>

long factorial(int n)
{
  if (n == 0)
    return 1;
  else
    return(n * factorial(n-1));
}

void main()
{
  int number;
  long fact;
```

```
15.  printf("Enter a number: ");
16.  scanf("%d", &number);
17.
18.  fact = factorial(number);
19.  printf("Factorial of %d is %ld\n", number, fact);
20.  return 0;
21.}
```

7. Armstrong Number

Code:

```
1.  #include<stdio.h>
2.   int main()
3.  {
4.  int n,r,sum=0,temp;
5.  printf("enter the number=");
6.  scanf("%d",&n);
7.  temp=n;
8.  while(n>0)
9.  {
10. r=n%10;
11. sum=sum+(r*r*r);
12. n=n/10;
13. }
14. if(temp==sum)
15. printf("armstrong  number ");
16. else
17. printf("not armstrong number");
18. return 0;
19. }
```

8. Sum of Digits

Code:

```
#include<stdio.h>
 int main()
{
int n,sum=0,m;
printf("Enter a number:");
scanf("%d",&n);
while(n>0)
{
m=n%10;
sum=sum+m;
n=n/10;
}
printf("Sum is=%d",sum);
return 0;
}
```

9. Reverse a Number

Code:

```
#include<stdio.h>
 int main()
{
int n, reverse=0, rem;
printf("Enter a number: ");
  scanf("%d", &n);
  while(n!=0)
  {
    rem=n%10;
```

```
    reverse=reverse*10+rem;
    n/=10;
  }
  printf("Reversed Number: %d",reverse);
return 0;
}
```

10. Swap two numbers without using third variable

Code:

```
 int main()
{
int a=10, b=20;
printf("Before swap a=%d b=%d",a,b);
a=a+b;//a=30 (10+20)
b=a-b;//b=10 (30-20)
a=a-b;//a=20 (30-10)
printf("\nAfter swap a=%d b=%d",a,b);
return 0;
}
```

11. Print the below pattern

```
    A
   ABA
  ABCBA
 ABCDCBA
ABCDEDCBA
```

Code:

```
#include<stdio.h>
#include<stdlib.h>
int main(){
 int ch=65;
  int i,j,k,m;
 system("cls");
  for(i=1;i<=5;i++)
  {
     for(j=5;j>=i;j--)
        printf(" ");
     for(k=1;k<=i;k++)
        printf("%c",ch++);
        ch--;
     for(m=1;m<i;m++)
        printf("%c",--ch);
     printf("\n");
     ch=65;
 }
return 0;
}
```

12. Sort an array

Code:

```
#include<stdio.h>
void main ()
{
  int i, j,temp;
  int a[10] = { 10, 9, 7, 101, 23, 44, 12, 78, 34, 23};
```

```
6.      for(i = 0; i<10; i++)
7.      {
8.          for(j = i+1; j<10; j++)
9.          {
10.             if(a[j] > a[i])
11.             {
12.                 temp = a[i];
13.                 a[i] = a[j];
14.                 a[j] = temp;
15.             }
16.         }
17.     }
18.     printf("Printing Sorted Element List ...\n");
19.     for(i = 0; i<10; i++)
20.     {
21.         printf("%d\n",a[i]);
22.     }
23.}
```

13. Program to print the largest and second largest element of the array.

Code:

```
1.  #include<stdio.h>
2.  void main ()
3.  {
4.      int arr[100],i,n,largest,sec_largest;
5.      printf("Enter the size of the array?");
6.      scanf("%d",&n);
7.      printf("Enter the elements of the array?");
8.      for(i = 0; i<n; i++)
```

```
    {
        scanf("%d",&arr[i]);
    }
    largest = arr[0];
    sec_largest = arr[1];
    for(i=0;i<n;i++)
    {
        if(arr[i]>largest)
        {
            sec_largest = largest;
            largest = arr[i];
        }
        else if (arr[i]>sec_largest && arr[i]!=largest)
        {
            sec_largest=arr[i];
        }
    }
    printf("largest = %d, second largest = %d",largest,sec_largest);

}
```

14. Leap year

Code:

```
#include<stdio.h>
#include<conio.h>
void main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if(((year%4==0) && ((year%400==0) || (year%100!==0))
```

```
8.      {
9.          printf("%d is a leap year", &year);
10.     } else {
11.         printf("%d is not a leap year", &year);
12.     }
13.     getch();
14.}
```

15. Perfect Number

Code:

```
1.  #include<stdio.h>
2.  #include<conio.h>
3.  void main()
4.  {
5.  // declare and initialize the variables
6.  int num, rem, sum = 0, i;
7.  // take an input from the user.
8.  printf("Enter a number\n");
9.  scanf("%d", &num);
10. // find all divisors and add them
11. for(i = 1; i < num; i++)
12.                 {
13.                         rem = num % i;
14.                         if (rem == 0)
15.                                 {
16.                                     sum = sum + i;
17.                                 }
18.                 }
19. if (sum == num)
20.                 printf(" %d is a Perfect Number");
```

```
21.         else
22.                 printf("\n %d is not a Perfect Number");
23. getch();
24. }
```

16. Find roots of QE

Code:

```
1.  #include<stdio.h>
2.  #include<math.h>  // it is used for math calculation
3.  #include<conio.h>
4.  void main()
5.  {
6.     float x, y, z, det, root1, root2, real, img;
7.     printf("\n Enter the value of coefficient x, y and z: \n ");
8.     scanf("%f %f %f", &x, &y, &z);
9.     // define the quadratic formula of the nature of the root
10.    det = y * y - 4 * x * z;
11.    // defines the conditions for real and different roots of the quadratic
    equation
12.    if (det > 0)
13.    {
14.    root1 = (-y + sqrt(det)) / (2 * x);
15.    root2 = (-y + sqrt(det)) / (2 * x);
16.    printf("\n Value of root1 = %.2f and value of root2 = %.2f", root1, root2);
17.    }
18.    // elseif condition defines both roots ( real and equal root) are equal in the
    quadratic equation
19.    else if (det == 0)
20.    {
21.       root1 = root2 = -y / (2 * x); // both roots are equal;
```

```
22.       printf("\n Value of root1 = %.2f and Value of root2 = %.2f", root1, root2);
23.   }
24.   // if det < 0, means both roots are real and imaginary in the quadratic
      equation.
25.   else {
26.       real = -y / (2 * x);
27.       img = sqrt(-det) / (2 * x);
28.       printf("\n value of root1 = %.2f + %.2fi and value of root2 = %.2f - %.2fi ",
      real, img, real, img);
29.   }
30.   getch();
31.   }
```

17. LCM of two numbers

Code:

```
1.  #include <stdio.h>
2.  #include <conio.h>
3.  void main()
4.  {
5.     int num1, num2, max_div, flag = 1;
6.     // accept any two positive number from the user
7.     printf( " Enter any two positive numbers to get the LCM \n ");
8.     scanf(" %d %d", &num1, &num2);
9.
10.    // max_div variable holds the max divisible number between num1 and
       num2.
11.    max_div = (num1 > num2) ? num1 : num2;
12.
13.    while (flag) // (flag = 1)
14.    {
```

```
15.      if (max_div % num1 == 0 && max_div % num2 == 0)
16.      {
17.         printf( " The LCM of %d and %d is %d. ", num1, num2, max_div);
18.         break;
19.      }
20.      ++max_div; // pre-increment max_div
21.   }
22.}
```

18. HCF of two numbers

Code:

```
1.  #include <stdio.h>
2.  #include <conio.h>
3.  int main()
4.  {
5.     // declare the variables
6.     int n1, n2, i, GCD_Num;
7.     printf ( " Enter any two numbers: \n ");
8.     scanf ( "%d %d", &n1, &n2);
9.
10.    // use for loop
11.    for( i = 1; i <= n1 && i <= n2; ++i)
12.    {
13.       if (n1 % i ==0 && n2 % i == 0)
14.          GCD_Num = i; /* if n1 and n2 is completely divisible by i, the divisible
       number will be the GCD_Num */
15.    }
16.    // print the GCD of two numbers
17.    printf (" GCD of two numbers %d and %d is %d.", n1, n2, GCD_Num);
18.    return 0;
19.}
```

19. Remove duplicates from an array

Code:

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    // declare local variables
    int arr[20], i, j, k, size;

    printf (" Define the number of elements in an array: ");
    scanf (" %d", &size);

    printf (" \n Enter %d elements of an array: \n ", size);
    // use for loop to enter the elements one by one in an array
    for ( i = 0; i < size; i++)
    {
        scanf (" %d", &arr[i]);
    }


    // use nested for loop to find the duplicate elements in array
    for ( i = 0; i < size; i ++)
    {
        for ( j = i + 1; j < size; j++)
        {
            // use if statement to check duplicate element
            if ( arr[i] == arr[j])
            {
                // delete the current position of the duplicate element
```

```
            for ( k = j; k < size - 1; k++)
            {
               arr[k] = arr [k + 1];
            }
            // decrease the size of array after removing duplicate element
            size--;

         // if the position of the elements is changes, don't increase the index j
            j--;
         }
      }
   }


   /* display an array after deletion or removing of the duplicate elements */
   printf (" \n Array elements after deletion of the duplicate elements: ");

   // for loop to print the array
   for ( i = 0; i < size; i++)
   {
      printf (" %d \t", arr[i]);
   }
   return 0;
}
```

20. Twin Primes

Code:

```
1. #include <stdio.h>
2. #include <conio.h>
3. int main ()
```

```
4.  {
5.      // declare variables
6.      int i, num, count = 0;
7.      printf (" Enter the last number: ");
8.      scanf (" %d", &num); // get the last number
9.
10.     for (i = 2; i <= num; i++)
11.     {
12.         if (twinprime (i) && twinprime (i+2))
13.         {
14.             printf (" \n The twin prime number is: (%d, %d) ", i, i+2);
15.             count++; // counter increment by 1
16.         }
17.     }
18.     printf (" \n \n Total number of twin prime pairs: %d", count);
19.     return 0;
20. }
21. // function definition
22. int twinprime( int n)
23. {
24.     int i = 2;
25.     // use for loop to find the twin prime
26.     for (i = 2; i <= n/2; i++)
27.     {
28.         // if n is completely divisible by 1 without leaving any remainder, it returns
    0
29.         if (n%i == 0)
30.             return 0;
31.     }
32.     // otherwise it returns 1
33.     if (i > n / 2)
```

```
34.       return 1;
35.}
```

21. String Anagrams

Code:

```
1.  #include <stdio.h>
2.
3.  // function definition
4.  int get_anagrm (char [],char []);
5.
6.
7.  int main ()
8.  {
9.      // declaration of the array
10.     char arr1 [50], arr2 [50];
11.     int count;
12.
13.     printf (" Enter the first string: \n ");
14.     scanf (" %s", arr1);
15.
16.
17.     printf (" Enter the second string: \n ");
18.     scanf (" %s", arr2);
19.
20.     // call function
21.     count = get_anagrm (arr1, arr2);
22.
23.     // use if-else statement to validate both strings are anagram or not.
24.
25.     if (count == 1)
```

```
26.    {
27.        printf (" %s and %s strings are an anagram of each other. \n", arr1, arr2);
28.    }
29.    else
30.    {
31.        printf (" %s and %s strings are not an anagram of each other. \n", arr1,
    arr2);
32.    }
33.
34.    return 0;
35.}
36.
37.
38.// function defnition
39.int get_anagrm (char arr1[], char arr2[])
40.{
41.    // create two num arrays and initialize their value as 0
42.    int num1[20] = {0}, num2[20] = {0}, i = 0;
43.
44.    // use while loop to check arr1 is not null
45.    while (arr1[i] != '\0')
46.    {
47.        num1[arr1[i] -'a']++;
48.        i++;
49.    }
50.
51.    i = 0;
52.
53.    // use while loop to check arr2 is not null
54.    while (arr2[i] != '\0')
55.    {
```

```
56.       num2[arr2[i] -'a']++;
57.       i++;
58.    }
59.
60.    for ( i = 0;i <20; i++)
61.    {
62.       if ( num1[i] != num2[i])
63.          return 0;
64.    }
65.    return 1;
66.}
```

22. Find the occurrence of substring in main string

Code:

```
1.  # include < stdio . h >
2.  # include < string . h >
3.  # include < stdlib >
4.  # include < conio . h >
5.  char str [ 100 ] , sub [ 100 ] ;
6.  int count = 0 , count1 = 0 ;
7.  void main()
8.  {
9.     int i , j , l , l1 , l2 ;
10.    printf ( " \ n enter a string : " ) ;
11.    scanf ( " % [ ^ \ n ] s " , str ) ;
12.    l1 = strlen ( str ) ;
13.    printf ( " \ n enter a substring : " ) ;
14.    scanf ( " % [ ^ \ n ] s " , sub ) ;
15.    l2 = strlen ( sub ) ;
16.    for ( i = 0 ; i < l1 ; )
```

```
{
    j = 0 ;
    count = 0 ;
    while ( ( str [ i ] = = sub [ j ] ) )
    {
        count + + ;
        i + + ;
        j + + ;
    }
    if ( count = = l2 )
    {
        count1 + + ;
        count = 0 ;
    }
    else
        i + + ;
}
printf ( " % s occurs % d times in % s " , sub , count1 , str ) ;
}
```

23. Binary Search

Code:

```
#include <stdio.h>

int binary_search(int arr[], int left, int right, int target) {
    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] == target) {
            return mid;
```

```
        } else if (arr[mid] < target) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    return -1;  // Target not found
}

int main() {
    int arr[] = {1, 3, 5, 7, 9};
    int n = sizeof(arr) / sizeof(arr[0]);
    int target = 5;

    int index = binary_search(arr, 0, n - 1, target);

    if (index == -1) {
        printf("Target not found\n");
    } else {
        printf("Target found at index %d\n", index);
    }

    return 0;
}
```