1] Frequently asked c programs in interview

Find out the perfect number using c program

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code 1:

1.**C program to check perfect number**

#include<stdio.h>

int main(){

  int n,i=1,sum=0;

  printf("Enter a number: ");

  scanf("%d",&n);

  while(i<n){

      if(n%i==0)

           sum=sum+i;

          i++;

  }

  if(sum==n)

      printf("%d is a perfect number",i);

  else

      printf("%d is not a perfect number",i);

  return 0;

}

Sample output:

Enter a number: 6

6 is a perfect number

Code 2:

1.**C program to find perfect numbers**

2.**C perfect number code**

3.**Perfect number program in c language**

#include<stdio.h>

int main(){

  int n,i,sum;

  int min,max;

  printf("Enter the minimum range: ");

  scanf("%d",&min);

  printf("Enter the maximum range: ");

  scanf("%d",&max);

  printf("Perfect numbers in given range is: ");

  for(n=min;n<=max;n++){

    i=1;

    sum = 0;

    while(i<n){

      if(n%i==0)

           sum=sum+i;

          i++;

    }

    if(sum==n)

      printf("%d ",n);

  }

  return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 20

Perfect numbers in given range is: 6

Code 3:

3.**C program to print perfect numbers from 1 to 100**

#include<stdio.h>

int main(){

  int n,i,sum;

  printf("Perfect numbers are: ");

  for(n=1;n<=100;n++){

    i=1;

    sum = 0;

    while(i<n){

      if(n%i==0)

           sum=sum+i;

          i++;

    }

    if(sum==n)

      printf("%d ",n);

  }

  return 0;

}

Output:

Perfect numbers are: 6 28

**Definition of perfect number or What is perfect number?**

Perfect number is a positive number which sum of all positive divisors excluding that number is equal to that number. For example 6 is perfect number since divisor of 6 are 1, 2 and 3.  Sum of its divisor is

1 + 2+ 3 =6

Note: 6 is the smallest perfect number.

Next perfect number is 28 since 1+ 2 + 4 + 7 + 14 = 28

Some more perfect numbers: 496, 8128

### \*\*\*\*Check the given number is armstrong number or not using c program

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Code 1:

1.**Warp to check a number is Armstrong**

2.**C program to check whether a number is Armstrong or not**

3.**Simple c program for Armstrong number**

4.**Armstrong number in c with output**

#include<stdio.h>

int main(){

    int num,r,sum=0,temp;

    printf("Enter a number: ");

    scanf("%d",&num);

    temp=num;

    while(num!=0){

         r=num%10;

         num=num/10;

         sum=sum+(r\*r\*r);

    }

    if(sum==temp)

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

    else

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

    return 0;

}

Sample output:

Enter a number: 153

153 is an Armstrong number

**The time complexity of a program that determines Armstrong number is**: O (Number of digits)

Code 2:

1.**Write a c program for Armstrong number**

2.**C program for Armstrong number generation**

3.**How to find Armstrong number in c**

4. **Code for Armstrong number in c**

#include<stdio.h>

int main(){

    int num,r,sum,temp;

    int min,max;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    printf("Armstrong numbers in given range are: ");

    for(num=min;num<=max;num++){

         temp=num;

         sum = 0;

         while(temp!=0){

             r=temp%10;

             temp=temp/10;

             sum=sum+(r\*r\*r);

         }

         if(sum==num)

             printf("%d ",num);

    }

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 200

Armstrong numbers in given range are: 1 153

Code 3:

1.**Armstrong number in c using for loop**

#include<stdio.h>

int main(){

    int num,r,sum=0,temp;

    printf("Enter a number: ");

    scanf("%d",&num);

    for(temp=num;num!=0;num=num/10){

         r=num%10;

         sum=sum+(r\*r\*r);

    }

    if(sum==temp)

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

    else

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

    return 0;

}

Sample output:

Enter a number: 370

370 is an Armstrong number

Logic of Armstrong number in c

Code 4:

1.**C program to print Armstrong numbers from 1 to 500**

2.**C program for finding Armstrong numbers**

#include<stdio.h>

int main(){

    int num,r,sum,temp;

    for(num=1;num<=500;num++){

         temp=num;

         sum = 0;

         while(temp!=0){

             r=temp%10;

             temp=temp/10;

             sum=sum+(r\*r\*r);

         }

         if(sum==num)

             printf("%d ",num);

    }

    return 0;

}

Output:

1 153 370 371 407

**Definition of Armstrong number or what is an Armstrong number:**

**Definition according to c programming point of view:**

## THOSE NUMBERS WHICH SUM OF THE CUBE OF ITS DIGITS IS EQUAL TO THAT NUMBER ARE KNOWN AS ARMSTRONG NUMBERS. FOR EXAMPLE 153 SINCE 1^3 + 5^3 + 3^3 = 1+ 125 + 9 =153

## OTHER ARMSTRONG NUMBERS: 370,371,407 ETC.

## IN GENERAL DEFINITION:

## THOSE NUMBERS WHICH SUM OF ITS DIGITS TO POWER OF NUMBER OF ITS DIGITS IS EQUAL TO THAT NUMBER ARE KNOWN AS ARMSTRONG NUMBERS.

## EXAMPLE 1: 153

## TOTAL DIGITS IN 153 IS 3

## AND 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153

Example 2: 1634

Total digits in 1634 is 4

And 1^4 + 6^4 + 3^4 +4^4 = 1 + 1296 + 81 + 64 =1634

Examples of Armstrong numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725

/\* Check given number is prime number or not using c program \*/

**Definition of prime number:**

A natural number greater than one has not any other divisors except 1 and itself. In other word we can say which has only two divisors 1 and number itself. For example: 5

Their divisors are 1 and 5.

Note: 2 is only even prime number.

**Logic for prime number in c**

We will take a loop and divide number from 2 to number/2. If the number is not divisible by any of the numbers then we will print it as prime number.

**Example of prime numbers** : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199 etc.

Code 1:

1.**C program to determine prime number**

2.**Determining if a number is prime in c**

3.**C program to find given number is prime or not**

#include<stdio.h>

int main(){

    int num,i,count=0;

    printf("Enter a number: ");

    scanf("%d",&num);

    for(i=2;i<=num/2;i++){

        if(num%i==0){

         count++;

            break;

        }

    }

   if(count==0 && num!= 1)

        printf("%d is a prime number",num);

   else

      printf("%d is not a prime number",num);

   return 0;

}

Sample output:

Enter a number: 5

5 is a prime number

Code 2:

1**. C program for prime numbers between 1 to 100**

2.**How to find prime numbers from 1 to 100 in c**

3.**How to print prime numbers from 1 to 100 in c**

#include<stdio.h>

int main(){

    int num,i,count;

    for(num = 1;num<=100;num++){

         count = 0;

         for(i=2;i<=num/2;i++){

             if(num%i==0){

                 count++;

                 break;

             }

        }

         if(count==0 && num!= 1)

             printf("%d ",num);

    }

   return 0;

}

Output:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Code 3:

1.**C program for prime numbers between 1 to n**

2.**C program to find prime numbers up to n**

3.**C program to list prime numbers**

4.**Write a c program to generate n prime numbers**

5.**C program to find n prime numbers**

#include<stdio.h>

int main(){

    int num,i,count,n;

    printf("Enter max range: ");

    scanf("%d",&n);

    for(num = 1;num<=n;num++){

         count = 0;

         for(i=2;i<=num/2;i++){

             if(num%i==0){

                 count++;

                 break;

             }

        }

         if(count==0 && num!= 1)

             printf("%d ",num);

    }

   return 0;

}

Sample output:

Enter max range: 50

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

Code 4:

1.**C program to find prime numbers using while loop**

2.**Wap to find prime numbers in c**

3.**Write a c program to generate prime number**

4.**How to get prime numbers in c**

#include<stdio.h>

int main(){

   int num,i,count,min,max;

printf("Enter min range: ");

    scanf("%d",&min);

    printf("Enter max range: ");

    scanf("%d",&max);

    num = min;

    while(num<=max){

         count = 0;

         i=2;

         while(i<=num/2){

             if(num%i==0){

                 count++;

                 break;

             }

             i++;

        }

         if(count==0 && num!= 1)

             printf("%d ",num);

         num++;

    }

   return 0;

}

Sample output:

Enter min range: 50

Enter max range: 100

53 59 61 67 71 73 79 83 89 97

Code 5:

1.**How to find out prime numbers in c programming**

2.**Display prime numbers in c**

3.**C program to find prime numbers between two numbers**

4.**C code to display prime numbers within a range**

#include<stdio.h>

int main(){

    int num,i,count,min,max;

     printf("Enter min range: ");

     scanf("%d",&min);

    printf("Enter max range: ");

    scanf("%d",&max);

    for(num = min;num<=max;num++){

         count = 0;

         for(i=2;i<=num/2;i++){

             if(num%i==0){

                 count++;

                 break;

             }

        }

         if(count==0 && num!= 1)

             printf("%d ",num);

    }

   return 0;

}

Sample output:

Enter min range: 10

Enter max range: 50

11 13 17 19 23 29 31 37 41 43 47

Code 6:

1. **Sum of prime numbers from 1 to 100 in c**

#include<stdio.h>

int main(){

    int num,i,count,sum=0;

    for(num = 1;num<=100;num++){

         count = 0;

         for(i=2;i<=num/2;i++){

             if(num%i==0){

                 count++;

                 break;

             }

        }

         if(count==0 && num!= 1)

             sum = sum + num;

    }

    printf("Sum of prime numbers is: %d ",sum);

   return 0;

}

Output:

Sum of prime numbers is: 1060

Code 7:

**1. C program to find sum of prime numbers**

#include<stdio.h>

int main(){

    int num,i,count,min,max,sum=0;

     printf("Enter min range: ");

     scanf("%d",&min);

    printf("Enter max range: ");

    scanf("%d",&max);

    for(num = min;num<=max;num++){

         count = 0;

         for(i=2;i<=num/2;i++){

             if(num%i==0){

                 count++;

                 break;

             }

        }

         if(count==0 && num!= 1)

             sum = sum + num;

    }

    printf("Sum of prime numbers is: %d ",sum);

   return 0;

}

Sample output:

Enter min range: 50

Enter max range: 100

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Write a c program to check given number is strong number or not.

Code 1:

1.**Write a c program to check whether a number is strong or not**

#include<stdio.h>

int main(){

  int num,i,f,r,sum=0,temp;

  printf("Enter a number: ");

  scanf("%d",&num);

  temp=num;

  while(num){

      i=1,f=1;

      r=num%10;

      while(i<=r){

         f=f\*i;

        i++;

      }

      sum=sum+f;

      num=num/10;

  }

  if(sum==temp)

      printf("%d is a strong number",temp);

  else

      printf("%d is not a strong number",temp);

  return 0;

}

Sample output:

Enter a number: 145

145 is a strong number

Code 2:

1.**C program for strong number**

2.**Strong number program in c**

#include<stdio.h>

int main(){

  int num,i,f,r,sum,temp;

  int min,max;

  printf("Enter minimum range: ");

  scanf("%d",&min);

  printf("Enter maximum range: ");

  scanf("%d",&max);

  printf("Strong numbers in given range are: ");

  for(num=min; num <= max; num++){

      temp = num;

      sum=0;

      while(temp){

           i=1;

           f=1;

           r=temp%10;

           while(i<=r){

             f=f\*i;

             i++;

           }

         sum=sum+f;

         temp=temp/10;

      }

      if(sum==num)

           printf("%d ",num);

  }

  return 0;

}

Sample output:

Enter minimum range: 100

Enter maximum range: 100000

Strong numbers in given range are: 145 40585

**Definition of strong number:**

A number is called strong number if sum of the factorial of its digit is equal to number itself. For example: 145 since

1! + 4! + 5! = 1 + 24 + 120 = 145

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

C program for odd or even number

Code 1:

1. **C program to check even or odd**

2. **C determine odd or even**

3. **How to check odd number in c**

4. **How to determine odd or even in c**

5. **C even odd test**

#include<stdio.h>

int main(){

    int number;

    printf("Enter any integer: ");

    scanf("%d",&number);

    if(number % 2 ==0)

         printf("%d is even number.",number);

    else

         printf("%d is odd number.",number);

    return 0;

}

Sample output:

Enter any integer: 5

5 is odd number.

Code 2:

1. **Display odd numbers in c**

2. **How to print odd numbers in c**

#include<stdio.h>

int main(){

    int number;

    int min,max;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    printf("Odd numbers in given range are: ");

    for(number = min;number <= max; number++)

         if(number % 2 !=0)

             printf("%d ",number);

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 20

Odd numbers in given ranges are: 1 3 5 7 9 11 13 15 17 19

Code 3:

1. **Even and odd numbers program in c**

2. **C program to find even or odd**

#include<stdio.h>

int main(){

    int number;

    int min,max;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    printf("Odd numbers in given range are: ");

    for(number = min;number <= max; number++)

         if(number % 2 !=0)

             printf("%d ",number);

    printf("\nEven numbers in given range are: ");

    for(number = min;number <= max; number++)

         if(number % 2 ==0)

             printf("%d ",number);

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 20

Odd numbers in given ranges are: 1 3 5 7 9 11 13 15 17 19

Even numbers in given ranges are: 2 4 6 8 10 12 14 16 18 20

Code 4:

1. **Sum of odd numbers in c**

#include<stdio.h>

int main(){

    int number;

    int min,max;

    long sum =0;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    for(number = min;number <= max; number++)

         if(number % 2 !=0)

             sum = sum + number;

    printf("Sum of odd numbers in given range is: %ld",sum);

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 100

Sum of odd numbers in given range is: 2500

Code 5:

1. **Sum of odd and even numbers c program**

#include<stdio.h>

int main(){

    int number;

    int min,max;

    long odd\_sum =0,even\_sum = 0;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    for(number = min;number <= max; number++)

         if(number % 2 != 0)

             odd\_sum = odd\_sum + number;

         else

             even\_sum = even\_sum + number;

    printf("Sum of even numbers in given range is: %ld\n",even\_sum);

    printf("Sum of odd numbers in given range is: %ld",odd\_sum);

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 10

Sum of even numbers in given range is: 30

Sum of odd numbers in given range is: 25

**Algorithm:**

Number is called even number if it is divisible by two otherwise odd.

Example of even numbers: 0,2,4,8,9,10 etc.

Example of odd numbers: 1, 3,5,7,9 etc.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Check the given number is palindrome number or not using c program

Code 1:

1. **Wap to check a number is palindrome**

2. **C program to find whether a number is palindrome or** **not**

#include<stdio.h>

int main(){

    int num,r,sum=0,temp;

    printf("Enter a number: ");

    scanf("%d",&num);

    temp=num;

    while(num){

         r=num%10;

         num=num/10;

         sum=sum\*10+r;

    }

    if(temp==sum)

         printf("%d is a palindrome",temp);

    else

         printf("%d is not a palindrome",temp);

    return 0;

}

Sample output:

Enter a number: 131

131 is a palindrome

Code 2:

1.**Write a c program for palindrome**

2.**C program to find palindrome of a number**

3.**Palindrome number in c language**

#include<stdio.h>

int main(){

    int num,r,sum,temp;

    int min,max;

    printf("Enter the minimum range: ");

    scanf("%d",&min);

    printf("Enter the maximum range: ");

    scanf("%d",&max);

    printf("Palindrome numbers in given range are: ");

    for(num=min;num<=max;num++){

         temp=num;

         sum=0;

         while(temp){

             r=temp%10;

             temp=temp/10;

             sum=sum\*10+r;

         }

         if(num==sum)

             printf("%d ",num);

    }

    return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 50

Palindrome numbers in given range are: 1 2 3 4 5 6 7 8 9 11 22 33 44

Code 3:

1.**How to check if a number is a palindrome using for loop**

#include<stdio.h>

int main(){

    int num,r,sum=0,temp;

    printf("Enter a number: ");

    scanf("%d",&num);

    for(temp=num;num!=0;num=num/10){

         r=num%10;

         sum=sum\*10+r;

    }

    if(temp==sum)

         printf("%d is a palindrome",temp);

    else

         printf("%d is not a palindrome",temp);

    return 0;

}

Sample output:

Enter a number: 1221

1221 is a palindrome

Code 4:

1. **C program to check if a number is palindrome using recursion**

#include<stdio.h>

int checkPalindrome(int);

int main(){

    int num,sum;

    printf("Enter a number: ");

    scanf("%d",&num);

    sum = checkPalindrome(num);

    if(num==sum)

         printf("%d is a palindrome",num);

    else

    printf("%d is not a palindrome",num);

    return 0;

}

int checkPalindrome(int num){

    static int sum=0,r;

    if(num!=0){

         r=num%10;

         sum=sum\*10+r;

         checkPalindrome(num/10);

    }

    return sum;

}

Sample output:

Enter a number: 25

25 is not a palindrome

**Definition of Palindrome number or What is palindrome number?**

A number is called palindrome number if it is remain same when its digits are reversed. For example 121 is palindrome number. When we will reverse its digit it will remain same number i.e. 121

**Palindrome numbers examples**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191 etc.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

C program for solving quadratic equation

1. **C program to calculate roots of a quadratic equation**

2. **Quadratic equation in c language**

#include<stdio.h>

#include<math.h>

int main(){

  float a,b,c;

  float d,root1,root2;

  printf("Enter a, b and c of quadratic equation: ");

  scanf("%f%f%f",&a,&b,&c);

  d = b \* b - 4 \* a \* c;

  if(d < 0){

    printf("Roots are complex number.\n");

    printf("Roots of quadratic equation are: ");

    printf("%.3f%+.3fi",-b/(2\*a),sqrt(-d)/(2\*a));

    printf(", %.3f%+.3fi",-b/(2\*a),-sqrt(-d)/(2\*a));

    return 0;

  }

  else if(d==0){

   printf("Both roots are equal.\n");

   root1 = -b /(2\* a);

   printf("Root of quadratic equation is: %.3f ",root1);

   return 0;

  }

  else{

   printf("Roots are real numbers.\n");

   root1 = ( -b + sqrt(d)) / (2\* a);

   root2 = ( -b - sqrt(d)) / (2\* a);

   printf("Roots of quadratic equation are: %.3f , %.3f",root1,root2);

  }

  return 0;

}

Sample output:

Enter a, b and c of quadratic equation: 2 4 1

Roots are real numbers.

Roots of quadratic equation are: -0.293, -1.707

1. **How to find a b and c in a quadratic equation**

#include<stdio.h>

#include<math.h>

int main(){

  float a,b,c;

  float d,root1,root2;

  printf("Enter quadratic equation in the format ax^2+bx+c: ");

  scanf("%fx^2%fx%f",&a,&b,&c);

  d = b \* b - 4 \* a \* c;

  if(d < 0){

    printf("Roots are complex number.\n");

    return 0;

  }

   root1 = ( -b + sqrt(d)) / (2\* a);

   root2 = ( -b - sqrt(d)) / (2\* a);

   printf("Roots of quadratic equation are: %.3f , %.3f",root1,root2);

  return 0;

}

Sample output:

Enter quadratic equation in the format ax^2+bx+c: 2x^2+4x+-1

Roots of quadratic equation are: 0.000, -2.000

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

TO FIND FIBONACCI SERIES USING C PROGRAM

Code 1:

1.**Write a program to generate the Fibonacci series in**c

2.**Write a program to print Fibonacci series in c**

3.**Basic c programs Fibonacci series**

4.**How to print Fibonacci series in c**

5.**How to find Fibonacci series in c programming**

6.**Fibonacci series in c using for loop**

#include<stdio.h>

int main(){

    int k,r;

    long int i=0l,j=1,f;

    //Taking maximum numbers form user

    printf("Enter the number range:");

    scanf("%d",&r);

    printf("FIBONACCI SERIES: ");

    printf("%ld %ld",i,j); //printing firts two values.

    for(k=2;k<r;k++){

         f=i+j;

         i=j;

         j=f;

         printf(" %ld",j);

    }

    return 0;

}

Sample output:

Enter the number range: 15

FIBONACCI SERIES: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

Code 2:

1.**Fibonacci series using array in c**

2.**Fibonacci series program in c language**

3.**Source code of Fibonacci series in c**

4.**Wap to print Fibonacci series in c**

#include<stdio.h>

int main(){

    int i,range;

    long int arr[40];

    printf("Enter the number range: ");

    scanf("%d",&range);

    arr[0]=0;

    arr[1]=1;

    for(i=2;i<range;i++){

         arr[i] = arr[i-1] + arr[i-2];

    }

    printf("Fibonacci series is: ");

    for(i=0;i<range;i++)

         printf("%ld ",arr[i]);

    return 0;

}

Sample output:

Enter the number range: 20

Fibonacci series is: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

Code 3:

1.**Fibonacci series in c using while loop**

2.**C program to calculate Fibonacci series**

3.**C program to display Fibonacci series**

4.**Fibonacci series in c with explanation**

5.**C code to generate Fibonacci series**

#include<stdio.h>

int main(){

    int k=2,r;

    long int i=0l,j=1,f;

    printf("Enter the number range:");

    scanf("%d",&r);

    printf("Fibonacci series is: %ld %ld",i,j);

    while(k<r){

         f=i+j;

         i=j;

         j=f;

         printf(" %ld",j);

          k++;

    }

    return 0;

}

Sample output:

Enter the number range: 10

Fibonacci series is: 0 1 1 2 3 5 8 13 21 34

Code 4:

1.**Sum of Fibonacci series in c**

#include<stdio.h>

int main(){

    int k,r;

    long int i=0,j=1,f;

    long int sum = 1;

    printf("Enter the number range: ");

    scanf("%d",&r);

    for(k=2;k<r;k++){

         f=i+j;

         i=j;

         j=f;

         sum = sum + j;

    }

    printf("Sum of Fibonacci series is: %ld",sum);

    return 0;

}

Sample output:

Enter the number range: 4

Sum of Fibonacci series is: 4

**Algorithm:**

**What is Fibonacci series?**

**Logic of Fibonacci series**

**Definition of Fibonacci numbers:**

We assume first two Fibonacci are 0 and 1

A series of numbers in which each sequent number is sum of its two previous numbers is known as Fibonacci series and each numbers are called Fibonacci numbers. So Fibonacci numbers is

**Algorithm for Fibonacci series**

Fn = Fn-2+ Fn-1

**Example of Fibonacci series:**

0 , 1 ,1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55  ...

5 is Fibonacci number since sum of its two previous number i.e. 2 and 3 is 5

8 is Fibonacci number since sum of its two previous number i.e. 3 and 5 is 8 and so on.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

TO FIND FACTORIAL OF A NUMBER USING C PROGRAM

Code 1:

1.**C code for factorial of a number**

2.**C program to find the factorial of a given number**

3.**Factorial program in c using while loop**  
4.**Factorial program in c without using recursion**

#include<stdio.h>

int main(){

  int i=1,f=1,num;

  printf("Enter a number: ");

  scanf("%d",&num);

  while(i<=num){

      f=f\*i;

      i++;

  }

  printf("Factorial of %d is: %d",num,f);

  return 0;

}

Sample output:

Enter a number: 5

Factorial of 5 is: 120

Code 2:

1.**Factorial program in c using for loop**

2.**Simple factorial program in c**

3.**C program to calculate factorial**

#include<stdio.h>

int main(){

  int i,f=1,num;

  printf("Enter a number: ");

  scanf("%d",&num);

  for(i=1;i<=num;i++)

      f=f\*i;

  printf("Factorial of %d is: %d",num,f);

  return 0;

}

Code 3:

1.**Factorial program in c using pointers**

2.**How to calculate factorial in c**

3.**Factorial program in c language**

#include<stdio.h>

void findFactorial(int,int \*);

int main(){

  int i,factorial,num;

  printf("Enter a number: ");

  scanf("%d",&num);

  findFactorial(num,&factorial);

  printf("Factorial of %d is: %d",num,\*factorial);

  return 0;

}

void findFactorial(int num,int \*factorial){

    int i;

    \*factorial =1;

    for(i=1;i<=num;i++)

      \*factorial=\*factorial\*i;

}

Code 4:

1.**Factorial program in c using function**

2.**C program to find factorial of a number**

#include<stdio.h>

int findFactorial(int);

int main(){

  int i,factorial,num;

  printf("Enter a number: ");

  scanf("%d",&num);

  factorial = findFactorial(num);

  printf("Factorial of %d is: %d",num,factorial);

  return 0;

}

int findFactorial(int num){

    int i,f=1;

    for(i=1;i<=num;i++)

      f=f\*i;

     return f;

}

Sample output:

Enter a number: 8

Factorial of 8 is: 40320

Code 5:

1.**Factorial series in c**

#include<stdio.h>

int main(){

  long f=1;

  int i,num,min,max;

  printf("Enter the minimum range: ");

  scanf("%d",&min);

  printf("Enter the maximum range: ");

  scanf("%d",&max);

  printf("Factorial series in given range: ");

  for(num=min;num<=max;num++){

    f=1;

    for(i=1;i<=num;i++)

      f=f\*i;

    printf("%ld ",f);

  }

  return 0;

}

Sample output:

Enter the minimum range: 1

Enter the maximum range: 10

Factorial series in given range: 1 2 6 24 120 720 5040 40320 362880 3628800

**Algorithm:**

**Factorial value**

Factorial of number is defined as:

Factorial (n) = 1\*2\*3 … \* n

For example: Factorial of 5 = 1\*2\*3\*4\*5 = 120

Note: Factorial of zero = 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Write a c program for Floyd’s triangle.

**1. Write a c program to print Floyd’s triangle**

**2. C program to display Floyd’s triangle**

**3. How to print Floyd’s triangle in c**

#include<stdio.h>

int main(){

  int i,j,r,k=1;

  printf("Enter the range: ");

  scanf("%d",&r);

  printf("FLOYD'S TRIANGLE\n\n");

  for(i=1;i<=r;i++){

      for(j=1;j<=i;j++,k++)

           printf(" %d",k);

      printf("\n");

  }

  return 0;

}

Sample output:

Enter the range: 10

FLOYD'S TRIANGLE

 1

 2 3

 4 5 6

 7 8 9 10

 11 12 13 14 15

 16 17 18 19 20 21

 22 23 24 25 26 27 28

 29 30 31 32 33 34 35 36

 37 38 39 40 41 42 43 44 45

 46 47 48 49 50 51 52 53 54 55

**What is Floyd’s triangle?**

**Definition of floyd's triangle:**

Floyd's triangle is a right angled-triangle using the natural numbers. Examples of floyd's triangle:

Example 1:

1

2 3

4 5 6

7 8 9 10

Example 2:

1

2   3

4   5   6

7   8   9   10

11  12  13  14  15

16  17  18  19  20 21

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Write a c program to print Pascal triangle.

1.**Pascal triangle in c without using array**

2.**C code to print Pascal triangle**

3**. Simple c program for Pascal triangle**

4.**C program to generate Pascal triangle**

5.**Pascal triangle program in c language**

6.**C program to print Pascal triangle using for loop**

#include<stdio.h>

long fact(int);

int main(){

    int line,i,j;

    printf("Enter the no. of lines: ");

    scanf("%d",&line);

    for(i=0;i<line;i++){

         for(j=0;j<line-i-1;j++)

             printf(" ");

         for(j=0;j<=i;j++)

             printf("%ld ",fact(i)/(fact(j)\*fact(i-j)));

         printf("\n");

    }

    return 0;

}

long fact(int num){

    long f=1;

    int i=1;

    while(i<=num){

         f=f\*i;

         i++;

  }

  return f;

 }

**Sample output:**

Enter the no. of lines: 8

       1

      1 1

     1 2 1

    1 3 3 1

   1 4 6 4 1

  1 5 10 10 5 1

 1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

TO FIND MULTIPLICATION TABLE USING C PROGRAM

1.**Multiplication tables in c program**

2.**Write a c program to print multiplication table**

3.**Code for multiplication table in c**

4.**Multiplication table in c language**

5.**Write a c program to print multiplication table**

#include<stdio.h>

int main(){

  int r,i,j,k;

  printf("Enter the number range: ");

  scanf("%d",&r);

  for(i=1;i<=r;i++){

      for(j=1;j<=10;j++)

           printf("%d\*%d=%d ",i,j,i\*j);

      printf("\n");

  }

  return 0;

}

Sample Output:

Enter the number range: 5

1\*1=1 1\*2=2 1\*3=3 1\*4=4 1\*5=5 1\*6=6 1\*7=7 1\*8=8 1\*9=9 1\*10=10

2\*1=2 2\*2=4 2\*3=6 2\*4=8 2\*5=10 2\*6=12 2\*7=14 2\*8=16 2\*9=18 2\*10=20

3\*1=3 3\*2=6 3\*3=9 3\*4=12 3\*5=15 3\*6=18 3\*7=21 3\*8=24 3\*9=27 3\*10=30

4\*1=4 4\*2=8 4\*3=12 4\*4=16 4\*5=20 4\*6=24 4\*7=28 4\*8=32 4\*9=36 4\*10=40

5\*1=5 5\*2=10 5\*3=15 5\*4=20 5\*5=25 5\*6=30 5\*7=35 5\*8=40 5\*9=45 5\*10=50

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PRINTING ASCII VALUE USING C PROGRAM

**Printing ascii value using c program**

**C code for ASCII table**

**C program to display ASCII values**

#include<stdio.h>

int main(){

    int i;

    for(i=0;i<=255;i++)

         printf("ASCII value of character %c: %d\n",i,i);

    return 0;

}

Output:

ASCII value of character  : 0

ASCII value of character ☺: 1

ASCII value of character ☻: 2

ASCII value of character ♥: 3

ASCII value of character ♦: 4

ASCII value of character ♣: 5

ASCII value of character ♠: 6

ASCII value of character : 7

ASCII value of character: 8

ASCII value of character        :

ASCII value of character

: 10

ASCII value of character ♂: 11

ASCII value of character ♀: 12

: 13I value of character

ASCII value of character ♫: 14

ASCII value of character ☼: 15

ASCII value of character ►: 16

ASCII value of character ◄: 17

ASCII value of character ↕: 18

ASCII value of character ‼: 19

ASCII value of character ¶: 20

ASCII value of character §: 21

ASCII value of character ▬: 22

ASCII value of character ↨: 23

ASCII value of character ↑: 24

ASCII value of character ↓: 25

ASCII value of character →: 26

ASCII value of character ←: 27

ASCII value of character ∟: 28

ASCII value of character ↔: 29

ASCII value of character ▲: 30

ASCII value of character ▼: 31

ASCII value of character  : 32

ASCII value of character !: 33

ASCII value of character ": 34

ASCII value of character #: 35

ASCII value of character $: 36

ASCII value of character %: 37

ASCII value of character &: 38

ASCII value of character ': 39

ASCII value of character (: 40

ASCII value of character ): 41

ASCII value of character \*: 42

ASCII value of character +: 43

ASCII value of character ,: 44

ASCII value of character -: 45

ASCII value of character .: 46

ASCII value of character /: 47

ASCII value of character 0: 48

ASCII value of character 1: 49

ASCII value of character 2: 50

ASCII value of character 3: 51

ASCII value of character 4: 52

ASCII value of character 5: 53

ASCII value of character 6: 54

ASCII value of character 7: 55

ASCII value of character 8: 56

ASCII value of character 9: 57

ASCII value of character :: 58

ASCII value of character ;: 59

ASCII value of character <: 60

ASCII value of character =: 61

ASCII value of character >: 62

ASCII value of character ?: 63

ASCII value of character @: 64

ASCII value of character A: 65

ASCII value of character B: 66

ASCII value of character C: 67

ASCII value of character D: 68

ASCII value of character E: 69

ASCII value of character F: 70

ASCII value of character G: 71

ASCII value of character H: 72

ASCII value of character I: 73

ASCII value of character J: 74

ASCII value of character K: 75

ASCII value of character L: 76

ASCII value of character M: 77

ASCII value of character N: 78

ASCII value of character O: 79

ASCII value of character P: 80

ASCII value of character Q: 81

ASCII value of character R: 82

ASCII value of character S: 83

ASCII value of character T: 84

ASCII value of character U: 85

ASCII value of character V: 86

ASCII value of character W: 87

ASCII value of character X: 88

ASCII value of character Y: 89

ASCII value of character Z: 90

ASCII value of character [: 91

ASCII value of character \: 92

ASCII value of character ]: 93

ASCII value of character ^: 94

ASCII value of character \_: 95

ASCII value of character `: 96

ASCII value of character a: 97

ASCII value of character b: 98

ASCII value of character c: 99

ASCII value of character d: 100

ASCII value of character e: 101

ASCII value of character f: 102

ASCII value of character g: 103

ASCII value of character h: 104

ASCII value of character i: 105

ASCII value of character j: 106

ASCII value of character k: 107

ASCII value of character l: 108

ASCII value of character m: 109

ASCII value of character n: 110

ASCII value of character o: 111

ASCII value of character p: 112

ASCII value of character q: 113

ASCII value of character r: 114

ASCII value of character s: 115

ASCII value of character t: 116

ASCII value of character u: 117

ASCII value of character v: 118

ASCII value of character w: 119

ASCII value of character x: 120

ASCII value of character y: 121

ASCII value of character z: 122

ASCII value of character {: 123

ASCII value of character |: 124

ASCII value of character }: 125

ASCII value of character ~: 126

ASCII value of character ⌂: 127

ASCII value of character Ç: 128

ASCII value of character ü: 129

ASCII value of character é: 130

ASCII value of character â: 131

ASCII value of character ä: 132

ASCII value of character à: 133

ASCII value of character å: 134

ASCII value of character ç: 135

ASCII value of character ê: 136

ASCII value of character ë: 137

ASCII value of character è: 138

ASCII value of character ï: 139

ASCII value of character î: 140

ASCII value of character ì: 141

ASCII value of character Ä: 142

ASCII value of character Å: 143

ASCII value of character É: 144

ASCII value of character æ: 145

ASCII value of character Æ: 146

ASCII value of character ô: 147

ASCII value of character ö: 148

ASCII value of character ò: 149

ASCII value of character û: 150

ASCII value of character ù: 151

ASCII value of character ÿ: 152

ASCII value of character Ö: 153

ASCII value of character Ü: 154

ASCII value of character ¢: 155

ASCII value of character £: 156

ASCII value of character ¥: 157

ASCII value of character ₧: 158

ASCII value of character ƒ: 159

ASCII value of character á: 160

ASCII value of character í: 161

ASCII value of character ó: 162

ASCII value of character ú: 163

ASCII value of character ñ: 164

ASCII value of character Ñ: 165

ASCII value of character ª: 166

ASCII value of character º: 167

ASCII value of character ¿: 168

ASCII value of character ⌐: 169

ASCII value of character ¬: 170

ASCII value of character ½: 171

ASCII value of character ¼: 172

ASCII value of character ¡: 173

ASCII value of character «: 174

ASCII value of character »: 175

ASCII value of character ░: 176

ASCII value of character ▒: 177

ASCII value of character ▓: 178

ASCII value of character │: 179

ASCII value of character ┤: 180

ASCII value of character ╡: 181

ASCII value of character ╢: 182

ASCII value of character ╖: 183

ASCII value of character ╕: 184

ASCII value of character ╣: 185

ASCII value of character ║: 186

ASCII value of character ╗: 187

ASCII value of character ╝: 188

ASCII value of character ╜: 189

ASCII value of character ╛: 190

ASCII value of character ┐: 191

ASCII value of character └: 192

ASCII value of character ┴: 193

ASCII value of character ┬: 194

ASCII value of character ├: 195

ASCII value of character ─: 196

ASCII value of character ┼: 197

ASCII value of character ╞: 198

ASCII value of character ╟: 199

ASCII value of character ╚: 200

ASCII value of character ╔: 201

ASCII value of character ╩: 202

ASCII value of character ╦: 203

ASCII value of character ╠: 204

ASCII value of character ═: 205

ASCII value of character ╬: 206

ASCII value of character ╧: 207

ASCII value of character ╨: 208

ASCII value of character ╤: 209

ASCII value of character ╥: 210

ASCII value of character ╙: 211

ASCII value of character ╘: 212

ASCII value of character ╒: 213

ASCII value of character ╓: 214

ASCII value of character ╫: 215

ASCII value of character ╪: 216

ASCII value of character ┘: 217

ASCII value of character ┌: 218

ASCII value of character █: 219

ASCII value of character ▄: 220

ASCII value of character ▌: 221

ASCII value of character ▐: 222

ASCII value of character ▀: 223

ASCII value of character α: 224

ASCII value of character ß: 225

ASCII value of character Γ: 226

ASCII value of character π: 227

ASCII value of character Σ: 228

ASCII value of character σ: 229

ASCII value of character µ: 230

ASCII value of character τ: 231

ASCII value of character Φ: 232

ASCII value of character Θ: 233

ASCII value of character Ω: 234

ASCII value of character δ: 235

ASCII value of character ∞: 236

ASCII value of character φ: 237

ASCII value of character ε: 238

ASCII value of character ∩: 239

ASCII value of character ≡: 240

ASCII value of character ±: 241

ASCII value of character ≥: 242

ASCII value of character ≤: 243

ASCII value of character ⌠: 244

ASCII value of character ⌡: 245

ASCII value of character ÷: 246

ASCII value of character ≈: 247

ASCII value of character °: 248

ASCII value of character ∙: 249

ASCII value of character ·: 250

ASCII value of character √: 251

ASCII value of character ⁿ: 252

ASCII value of character ²: 253

ASCII value of character ■: 254

ASCII value of character  : 255

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

C program to print hello world without using semicolon

**C program to print hello world without using semicolon**

#include<stdio.h>

void main(){

    if(printf("Hello world")){

    }

}

Solution: 2

#include<stdio.h>

void main(){

    while(!printf("Hello world")){

    }

}

Solution: 3

#include<stdio.h>

void main(){

    switch(printf("Hello world")){

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

write a c program which produces its own source code as its output

**How do you write a program which produces its own source code as its output in c language?**

#include<stdio.h>

int main(){

    FILE \*fp;

    char c;

    fp = fopen(\_\_FILE\_\_,"r");

    do{

         c= getc(fp);

         putchar(c);

    }

    while(c!=EOF);

    fclose(fp);

    return 0;

}

Output:

#include<stdio.h>

int main(){

    FILE \*fp;

    char c;

    fp = fopen(\_\_FILE\_\_,"r");

    do{

         c= getc(fp);

         putchar(c);

    }

    while(c!=EOF);

    fclose(fp);

    return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/