

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

Code 4:

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 5:

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 6:

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 7:

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

**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 8:

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 9:

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 10:

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 11:

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 12:

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 13:

## 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 14:

## 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

```

Code 15:

1. Write a c program to reverse a given number
2. C program to find reverse of a number
3. C program to reverse the digits of a number
4. Reverse of a number in c using while loop

```

#include<stdio.h>
int main(){
    int num,r,reverse=0;

    printf("Enter any number: ");
    scanf("%d",&num);

    while(num){
        r=num%10;
        reverse=reverse*10+r;
        num=num/10;
    }

    printf("Reversed of number: %d",reverse);
    return 0;
}

```

Sample output:

```

Enter any number: 12

```

Reversed of number: 21

Code 16:

**1. Reverse very large or big numbers beyond the range of long int**

**2. Reverse five digit number c program**

Logic is we accept the number as string

```
#include<stdio.h>
#define MAX 1000

int main(){

    char num[MAX];
    int i=0,j,flag=0;

    printf("Enter any positive integer: ");
    scanf("%s",num);

    while(num[i]){
        if(num[i] < 48 || num[i] > 57){
            printf("Invalid integer number");
            return 0;
        }
        i++;
    }

    printf("Reverse: ");
    for(j=i-1;j>=0;j--){
        if(flag==0 && num[j] ==48){
        }
        else{
            printf("%c",num[j]);
            flag =1;
        }
    }

    return 0;
}
```

Sample output:

Enter any positive integer:

234561000045645679001237800000000000

Reverse: 8732100976546540000165432

Code 17:

1. C program to reverse a number using for loop
2. How to find reverse of a number in c
3. Wap to reverse a number in c

```
#include<stdio.h>
int main(){
    int num,r,reverse=0;

    printf("Enter any number: ");
    scanf("%d",&num);

    for(;num!=0;num=num/10){
        r=num%10;
        reverse=reverse*10+r;
    }

    printf("Reversed of number: %d",reverse);
    return 0;
}
```

Sample output:

Enter any number: 123

Reversed of number: 321

Code 18:

1. C program to reverse a number using recursion

```
#include<stdio.h>
int main(){
    int num,reverse;

    printf("Enter any number: ");
    scanf("%d",&num);

    reverse=rev(num);
    printf("Reverse of number: %d",reverse);
    return 0;
}
```

```

int rev(int num) {
    static sum, r;

    if (num) {
        r=num%10;
        sum=sum*10+r;
        rev(num/10);
    }
    else
        return 0;

    return sum;
}

```

Sample output:

Enter any number: 456

Reverse of number: 654

Code 19:

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 20:

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

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

Code 21:

1. C program to add digits of a number
2. C program for sum of digits of a number
3. C program to calculate sum of digits

```

#include<stdio.h>
int main(){
    int num,sum=0,r;
    printf("Enter a number: ");
    scanf("%d",&num);
    while(num){
        r=num%10;
        num=num/10;
    }
}

```

```
        sum=sum+r;
    }
    printf("Sum of digits of number:  %d",sum);
    return 0;
}
```

Sample output:

Enter a number: 123

Sum of digits of number: 6

Code 22:

### 1. Sum of digits of a number in c using for loop

```
#include<stdio.h>
int main() {
    int num,sum=0,r;
    printf("Enter a number: ");
    scanf("%d",&num);

    for (;num!=0;num=num/10) {
        r=num%10;
        sum=sum+r;
    }
    printf("Sum of digits of number:  %d",sum);
    return 0;
}
```

Sample output:

Enter a number: 567

Sum of digits of number: 18

Code 23:

### 1. Sum of digits in c using recursion

```
#include<stdio.h>

int getSum(int);
int main() {
    int num,sum;
    printf("Enter a number: ");
    scanf("%d",&num);
```

```

    sum = getSum(num);

    printf("Sum of digits of number:  %d", sum);
    return 0;
}

int getSum(int num) {

    static int sum =0, r;

    if(num!=0) {
        r=num%10;
        sum=sum+r;
        getSum(num/10);
    }

    return sum;
}

```

Sample output:

Enter a number: 45

Sum of digits of number: 9

---

Code 24:

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 25:

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 26:

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 27:

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.

Code 28:

### Extract digits from integer in c language

```
#include<stdio.h>
int main(){
    int num,temp,factor=1;

    printf("Enter a number: ");
    scanf("%d",&num);

    temp=num;
    while(temp){
        temp=temp/10;
        factor = factor*10;
    }

    printf("Each digits of given number are: ");
    while(factor>1){
        factor = factor/10;
        printf("%d ",num/factor);
        num = num % factor;
    }

    return 0;
}
```

Sample output:

Enter a number: 123

Each digits of given number are: 1 2 3



Code 29:

### Program in c to print 1 to 100 without using loop

```
include<stdio.h>

int main(){
    int num = 1;

    print(num);

    return 0;
}
int print(num){
    if(num<=100){
        printf("%d ",num);
        print(num+1);
    }
}
```

Output:

Sample output:

```
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 56 57
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
94 95 96 97 98 99 100
```

Code 30:

### How to pass one dimensional array to function in c

```
#include <stdio.h>
#define N 5
void fstore1D(int a[], int a_size);
void fretrieve1D(int a[], int a_size);
void fedit1D(int a[], int a_size);
int main(){
    int a[N];
```

```

        printf("Input data into the matrix:\n");
fstore1D(a, N);
    fretrieve1D(a, N);
    fedit1D(a, N);
    fretrieve1D(a, N);
    return 0;
}

void fstore1D(int a[], int n){
    int i;
    for ( i = 0; i < n; ++i )
        scanf("%d", &a[i]);
}

void fretrieve1D(int a[], int n){
    int i;
    for ( i = 0; i < n; ++i )
        printf("%6d ", a[i]);
    printf("\n");
}

void fedit1D(int a[], int n){
    int i, q;
    for ( i = 0; i < n; ++i ){
        printf("Prev. data: %d\nEnter 1 to edit 0 to
skip.", a[i]);
        scanf("%d", &q);
        if ( q == 1 ){
            printf("Enter new value: ");
            scanf("%d", &a[i]);
        }
    }
}

```

Code 31:

How to pass two dimensional array to a function in c

```

#include <stdio.h>
#define M 3
#define N 5
void fstore2D(int a[][N]);
void fretrieve2D(int a[][N]);
int main(){
    int a[M][N];
    printf("Input data in matrix (%d X %d)\n", M, N);
    fstore2D(a);
    fretrieve2D(a);
    return 0;
}
void fstore2D(int a[][N]){
    int i, j;
    for (i = 0; i < M; ++i){
        for (j = 0; j < N; ++j)
            scanf("%d", &a[i][j]);
    }
}
void fretrieve2D(int a[][N]){
    int i, j;
    for ( i = 0; i < M; ++i ){
        for ( j = 0; j < N; ++j)
            printf("%6d ", a[i][j]);
        printf("\n");
    }
}

```

Code 32:

**How to calculate power of a number in c**

**How to write power in c**

```

#include<stdio.h>
int main(){
    int pow,num,i=1;
    long int sum=1;
    printf("\nEnter a number: ");

```

```

scanf("%d",&num);
printf("\nEnter power: ");
scanf("%d",&pow);
while(i<=pow){
    sum=sum*num;
    i++;
}
printf("\n%d to the power %d is: %ld",num,pow,sum);
return 0;
}

```

### Code 33:

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>
int main(){
    int line,i,j,k;
    printf("Enter the no. of lines");
    scanf("%d",&line);
    for(i=1;i<=line;i++){
        for(j=1;j<=line-i;j++)
            printf(" ");
        for(k=1;k<i;k++)
            printf("%d",k);
        for(k=i;k>=1;k--)
            printf("%d",k);
        printf("\n");
    }
    return 0;
}

```

### Code 34:

**Write a c program to find largest among three numbers using binary minus operator**

```
#include<stdio.h>
int main() {
    int a,b,c;
    printf("\nEnter 3 numbers: ");
    scanf("%d %d %d",&a,&b,&c);
    if(a-b>0 && a-c>0)
        printf("\nGreatest is a :%d",a);
    else
        if(b-c>0)
            printf("\nGreatest is b :%d",b);
        else
            printf("\nGreatest is c :%d",c);
    return 0;
}
```

Code 35:

**Add two numbers in c without using operator**

**How to add two numbers without using the plus operator in c**

```
#include<stdio.h>

int main() {

    int a,b;
    int sum;

    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);

    //sum = a - (-b);
    sum = a - ~b -1;

    printf("Sum of two integers: %d",sum);
}
```

```
    return 0;
}
```

Code 36:

**Write a c program to subtract two numbers without using subtraction operator**

```
#include<stdio.h>
```

```
int main() {
```

```
    int a,b;
```

```
    int sum;
```

```
    printf("Enter any two integers: ");
```

```
    scanf("%d%d",&a,&b);
```

```
    sum = a + ~b + 1;
```

```
    printf("Difference of two integers: %d",sum);
```

```
    return 0;
```

```
}
```

**Sample Output:**

```
Enter any two integers: 5 4
```

```
Difference of two integers: 1
```

Code 37:

**Write a c program to find largest among three numbers using conditional operator**

```
#include<stdio.h>
```

```
int main() {
```

```
    int a,b,c,big;
```

```
    printf("\nEnter 3 numbers:");
```

```
    scanf("%d %d %d",&a,&b,&c);
```

```
    big=(a>b&&a>c?a:b>c?b:c);
```

```

    printf("\nThe biggest number is:%d",big);
    return 0;
}

```

Code 38:

Write a c program to find out NCR factor of given number

```

#include<stdio.h>
int main() {
    int n,r,ncr;
    printf("Enter any two numbers->");
    scanf("%d %d",&n,&r);
    ncr=fact(n)/(fact(r)*fact(n-r));
    printf("The NCR factor of %d and %d is %d",n,r,ncr);
    return 0;
}
int fact(int n) {
    int i=1;
    while(n!=0) {
        i=i*n;
        n--;
    }
    return i;
}

```

Code 39:

How to convert string to int without using library functions in c

```

#include<stdio.h>

int stringToInt(char[] );
int main() {

    char str[10];
    int intValue;

    printf("Enter any integer as a string: ");
    scanf("%s",str);

    intValue = stringToInt(str);
}

```

```

    printf("Equivalent integer value: %d",intValue);

    return 0;
}

int stringToInt(char str[]){
    int i=0,sum=0;

    while(str[i]!='\0'){
        if(str[i]< 48 || str[i] > 57){
            printf("Unable to convert it into
integer.\n");
            return 0;
        }
        else{
            sum = sum*10 + (str[i] - 48);
            i++;
        }
    }

    return sum;
}

```

Sample output:

```

Enter any integer as a string: 123
Equivalent integer value: 123

```

Code 40:

### C program for swapping of two numbers

#### Code for swapping in c

```

#include<stdio.h>
int main(){
    int a,b,temp;

    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);
    printf("Before swapping: a = %d, b=%d",a,b);

```



```
temp= a;
a=b;
b=a;
printf("\nAfter swapping: a = %d, b=%d",a,b);

return 0;
```

Code 41:

**C program for swapping of two numbers using pointers**

```
#include<stdio.h>

int main(){

    int a,b;
    int *ptra,*ptrb;
    int *temp;

    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);

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

    ptra = &a;
    ptrb = &b;

    temp = ptra;
    *ptra = *ptrb;
    *ptrb = *temp;

    printf("\nAfter swapping: a = %d, b=%d",a,b);
    return 0;
}
```

Sample output:

```
Enter any two integers: 5 10
Before swapping: a = 5, b=10
After swapping: a = 10, b=10
```

Code 42:

### Swapping program in c using function

```
#include<stdio.h>

void swap(int *,int *);
int main(){

    int a,b;

    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);

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

    swap(&a,&b);

    printf("\nAfter swapping: a = %d, b=%d",a,b);
    return 0;
}

void swap(int *a,int *b){
    int *temp;
    temp = a;
    *a=*b;
    *b=*temp;
}
```

Sample output:

```
Enter any two integers: 3 6
Before swapping: a = 3, b=6
After swapping: a = 6, b=6
```

Code 43:

**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
```

---

---

### **CHECKING LEAP YEAR USING C PROGRAM**

#### **Definition of leap year:**

Rule 1: A year is called leap year if it is divisible by 400.

For example: 1600, 2000 etc leap year while 1500, 1700 are not leap year.

Rule 2: If year is not divisible by 400 as well as 100 but it is divisible by 4 then that year are also leap year.

For example: 2004, 2008, 1012 are leap year.

### **Leap year logic or Algorithm of leap year or Condition for leap year:**

```
IF year MODULER 400 IS 0
    THEN leap_year
ELSE IF year MODULER 100 IS 0
    THEN not_leap_year
ELSE IF year MODULER 4 IS 0
    THEN leap_year
ELSE
    not_leap_year
```

Code 44:

1. **C program to determine leap year**
2. **C program to find leap year or not**
3. **Leap year calculation in c**

```
#include<stdio.h>

int main(){
    int year;

    printf("Enter any year: ");
    scanf("%d",&year);
```

```
    if(((year%4==0) && (year%100!=0)) || (year%400==0))
        printf("%d is a leap year",year);
    else
        printf("%d is not a leap year",year);
    return 0;
}
```

Sample output:

Enter any year: 2010

2010 is not a leap year

Code 45:

1. **Write a program to find leap year in c**

2. **How to find leap year in c code**

```
#include<stdio.h>
```

```
int main(){
```

```
    int year;
```

```
    int min_year,max_year;
```

```
    printf("Enter the lowest year: ");
```

```
    scanf("%d",&min_year);
```

```
    printf("Enter the heighest year: ");
```

```
    scanf("%d",&max_year);
```

```

printf("Leap years in given range are: ");
for(year = min_year;year <= max_year; year++){
    if(((year%4==0)&&(year%100!=0))|| (year%400==0)
)
        printf("%d ",year);
    }
return 0;
}

```

Sample output:

Enter the lowest year: 2000

Enter the highest year: 2011

Leap years in given range is: 2000 2004 2008

Code 45:

**Write a c program to find out L.C.M. of two numbers.**

**LCM program in c with two numbers :**

```

#include<stdio.h>
int main() {
    int n1,n2,x,y;
    printf("\nEnter two numbers:");
    scanf("%d %d",&n1,&n2);
    x=n1,y=n2;
    while(n1!=n2) {
        if(n1>n2)
            n1=n1-n2;
        else
            n2=n2-n1;
    }
}

```

```
printf("L.C.M=%d",x*y/n1);  
return 0;  
}
```

LCM program in c with two numbers (Other logic) :

```
#include<stdio.h>  
  
int lcm(int,int);  
  
int main() {  
  
    int a,b,l;  
    printf("Enter any two positive integers ");  
    scanf("%d%d",&a,&b);  
  
    if(a>b)  
        l = lcm(a,b);  
    else  
        l = lcm(b,a);  
  
    printf("LCM of two integers is %d",l);  
  
    return 0;  
}  
  
int lcm(int a,int b) {  
  
    int temp = a;  
  
    while(1) {  
        if(temp % b == 0 && temp % a == 0)  
            break;  
        temp++;  
    }  
  
    return temp;  
}
```



Code 46:

LCM program in c with multiple numbers :

```
#include<stdio.h>

int lcm(int,int);

int main() {

    int a,b=1;
    printf("Enter positive integers. To quit press
zero.");

    while(1) {
        scanf("%d",&a);
        if(a<1)
            break;
        else if(a>b)
            b = lcm(a,b);
        else
            b = lcm(b,a);
    }

    printf("LCM is %d",b);

    return 0;
}

int lcm(int a,int b) {

    int temp = a;

    while(1) {
        if(temp % b == 0 && temp % a == 0)
            break;
        temp++;
    }

    return temp;
}
```

---

**Definition of LCM (Least common multiple):**

LCM of two integers is a smallest positive integer which is multiple of both integers that it is divisible by the both of the numbers.

For example: LCM of two integers 2 and 5 is 10 since 10 is the smallest positive numbers which is divisible by both 2 and 5.

---

---

**Find g.c.d of two number using c program****Definition of HCF (Highest common factor):**

HFC is also called greatest common divisor (gcd). HCF of two numbers is a largest positive numbers which can divide both numbers without any remainder. For example HCF of two numbers 4 and 8 is 2 since 2 is the largest positive number which can dived 4 as well as 8 without a remainder.

**Logic of HCF or GCD of any two numbers:**

In HCF we try to find any largest number which can divide both the number.

For example: HCF or GCD of 20 and 30

Both number 20 and 30 are divisible by 1, 2,5,10.

HCF=max (1, 2, 3, 4, 10) =10

## Logic for writing program:

It is clear that any number is not divisible by greater than number itself. In case of more than one numbers, a possible maximum number which can divide all of the numbers must be minimum of all of that numbers.

For example: 10, 20, and 30

Min (10, 20, 30) =10 can divide all there numbers. So we will take one for loop which will start form min of the numbers and will stop the loop when it became one, since all numbers are divisible by one. Inside for loop we will write one if conditions which will check divisibility of both the numbers.

Program:

```
#include<stdio.h>
```

```
int main() {
```

```
    int x,y,m,i;
```

```
    printf("Insert any two number: ");
```

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

```

if (x>y)
    m=y;
else
    m=x;

for (i=m; i>=1; i--) {
    if (x%i==0&&y%i==0) {
        printf("\nHCF of two number is : %d", i) ;
        break;
    }
}

return 0;
}

```

Code 46:

**Other logic : HCF program with two numbers in c**

```

#include<stdio.h>
int main() {
    int n1, n2;
    printf("\nEnter two numbers:");
    scanf("%d %d", &n1, &n2);
    while (n1!=n2) {
        if (n1>=n2-1)

```

```
    n1=n1-n2;

    else

    n2=n2-n1;

}

printf("\nGCD=%d",n1);

return 0;

}
```

Code 47:

**HCF program with multiple numbers in c**

```
#include<stdio.h>
```

```
int main() {
```

```
    int x,y=-1;
```

```
    printf("Insert numbers. To exit insert zero: ");
```

```
    while(1) {
```

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

```
        if(x<1)
```

```
            break;
```

```
        else if (y == -1)
            y = x;
        else if (x < y)
            y = gcd(x, y);
        else
            y = gcd(y, x);
    }

    printf("GCD is %d", y);

    return 0;
}

int gcd(int x, int y) {
    int i;
    for (i = x; i >= 1; i--) {
        if (x % i == 0 && y % i == 0) {
            break;
        }
    }
    return i;
}
```

## Find g.c.d of two number using c program

### **Definition of HCF (Highest common factor):**

HFC is also called greatest common divisor (gcd). HCF of two numbers is a largest positive numbers which can divide both numbers without any remainder. For example HCF of two numbers 4 and 8 is 2 since 2 is the largest positive number which can dived 4 as well as 8 without a remainder.

### **Logic of HCF or GCD of any two numbers:**

In HCF we try to find any largest number which can divide both the number.

For example: HCF or GCD of 20 and 30

Both number 20 and 30 are divisible by 1, 2,5,10.

HCF=max (1, 2, 3, 4, 10) =10

### **Logic for writing program:**

It is clear that any number is not divisible by greater than number itself. In case of more than one numbers, a possible maximum number which can divide all of the numbers must be minimum of all of that numbers.

For example: 10, 20, and 30

Min (10, 20, 30) =10 can divide all there numbers. So we will take one for loop which will start form min of the numbers and will stop the loop when it became one, since all numbers are divisible by one. Inside for loop we will write one if conditions which will check divisibility of both the numbers.

Program:

```
#include<stdio.h>
```

```
int main () {
```

```
    int x,y,m,i;
```

```
    printf("Insert any two number: ");
```

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

```
    if(x>y)
```

```
        m=y;
```

```
    else
```

```
        m=x;
```

```
    for(i=m;i>=1;i--){
```

```
        if(x%i==0&&y%i==0){
```

```
            printf("\nHCF of two number is : %d",i) ;
```



```

        break;
    }
}
return 0;
}

```

Code 48:

**Other logic : HCF program with two numbers in c**

```

#include<stdio.h>
int main(){
    int n1,n2;
    printf("\nEnter two numbers:");
    scanf("%d %d",&n1,&n2);
    while(n1!=n2){
        if(n1>=n2-1)
            n1=n1-n2;
        else
            n2=n2-n1;
    }
    printf("\nGCD=%d",n1);
    return 0;
}

```

Code 49:

HCF program with multiple numbers in c

```
#include<stdio.h>

int main() {

    int x,y=-1;

    printf("Insert numbers. To exit insert zero: ");

    while(1) {

        scanf("%d",&x);

        if(x<1)

            break;

        else if(y==-1)

            y=x;

        else if (x<y)

            y=gcd(x,y);

        else

            y=gcd(y,x);

    }

    printf("GCD is %d",y);

    return 0;
```

```
}
```

```
int gcd(int x,int y) {  
    int i;  
    for(i=x;i>=1;i--) {  
        if(x%i==0&&y%i==0) {  
            break;  
        }  
    }  
    return i;  
}
```

Code 50:

GCD of 3 numbers with high efficiency

```
#include  
int gcd(int , int , int);  
int main()  
{  
    int i , j , k , g;  
    scanf("%d %d %d", &i , &j , &k);  
  
    g = gcd(i , j , k);  
    printf("%d",g);  
  
    return 0;  
}  
  
int gcd(int i , int j , int k)  
{
```

```
int least;
least = i;
while(!( (i == j) && (j == k) ))
{
i = (i == 0 ? least : i);
j = (j == 0 ? least : j);
k = (k == 0 ? least : k);
if(i <= j)
{
if(i <= k)
least = i;
else
least = k;
}
else
{
if(j <= k)
least = j;
else
least = k;
}
i = i % least;
j = j % least;
k = k % least;
}
return least;
}
```