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

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

```
#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
```

Write a c program to find perfect numbers from given range.

```
#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);

if(min >= max)
{
printf("Please enter proper range\n");
return 0;
}

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

Write a C program to check whether given number is Armstrong number or not.

Those numbers which sum of the cube of its digits is equal to that number are known as Armstrong numbers.
For example 153
(1^3) + (5^3) + (3^3) = 1+ 125 + 9 =153
Other Armstrong numbers are : 370,371,407 etc.

General Definition :
Those numbers which sum of its digits to power of number of ots digits is equal to that number are known as Armstrong numbers.

Example : 1634
Total digits in 1634 is 4
And (1^4) + (6^4) + (3^4) + (4^4) = 1 + 1296 + 81 + 64 =1634

```
#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

Write a C program to check whether a given number is Prime number or not.

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.

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.

```
#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

Write a C program to find all the Prime numbers from given 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);

if(min >= max)
{
printf("Please enter proper range\n");
return 0;
}

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

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

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

```
#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
```

Write a program to check whether given number is prime number or not.

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

```
#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

Write a program to find factorial of given number.

```
#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

Write a C program to print Fibonacci series.

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

```
int k,r;
int i=0,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

Write a C program to print hello world without using semicolon.

Solution: 1

```
#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.

```
#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;
}
```

Write a C program to reverse the number.

```
#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;
}
```

Write a c program to find out sum of digit of given number.

```
#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;
}
```

Write a C program to find power of a given number.

```
#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;
}
```

Write a c program to add two numbers without using addition operator.

```
#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;
}
```

logic:

If we have to add numbers a and b then it can be done as

sum = a - (-b);

sum = a - (~b + 1);

sum = a - ~b – 1;

Write a c program or code 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
```

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;
}
```

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;
}
```

Write a C program to find number of digits.

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

long int num,sum,r;

printf("\nEnter a number:-");
scanf("%ld",&num);

while(num > 10)
{
sum = 0;
```

```
while(num)
{
r = num % 10;
num = num / 10;
sum = sum + r;
}

if(sum > 10)
num = sum;

else

break;
}

printf("\nSum of the digits in single digit is: %ld",sum);

return 0;
}
```

Write a C program for calculation of generic root.

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

long int num,sum,r;
printf("\nEnter a number:-");
scanf("%ld",&num);

while(num > 10)
{
sum = 0;
while(num)
{
r = num % 10;
num = num / 10;
sum += r;
}
```

```
if(sum > 10)
{
num = sum;
}
else
{
break;
}
}

printf("\nSum of the digits in single digit is: %ld",sum);

return 0;
}
```

Generic Root:-

It sum of digits of a number unit we don't get a single digit. For example:
Generic root of 456: 4 + 5 + 6 = 15
since 15 is two digit numbers so 1 + 5 = 6
Generic root of 456 = 6

Write a C program for calculation of generic root in one line.

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

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

printf("Generic root: %d",(x=num%9)?x:9);

return 0;
}
```

Sample output:

Enter any number: 731

Generic root: 2

Write a C program Prime factor of a number.

```
#include<stdio.h>
int main()
{
int num,i=1,j,k;

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

while(i<=num)
{
k=0;
if(num%i==0)
{
j=1;
while(j<=i)
{
if(i%j==0)
{
k++;
}
j++;
}
if(k==2) //for 1 and number itse
{
printf("\n%d is a prime factor",i);
}
}
i++;
}

return 0;
}
```

Write a c program to find out NCR factor of given number OrC program to find the ncr value by using recursive function.

```
#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;
}
```

This is a code which finds the combination.
The ***Combinations*** is a method of selecting several items or symbols out of a larger group or a data set, where an order does not matter. The Combination represented by **nCr** and calculated from the formula
nCr = n!/(r!(n - r)!)
Each r combination can be arranged in r! different ways. Then the number of r-permutations is equal to the number of r combinations times r!

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);
}

}
```

Simple program of c find the largest number.

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

printf("Enter the values of n: ");
scanf("%d",&n);

printf("Number %d",1);
scanf("%d",&big);

for(i=2;i<=n;i++)
{
printf("Number %d: ",i);
scanf("%d",&num);

if(big<num)
{
```

```
big=num;
}

}

printf("Largest number is: %d",big);

return 0;
}
```

Sample Output:
Enter the values of n:
Number 1: 12
Number 2: 32
Number 3: 35
Largest number is: 35

Count the number of digits in c programming language.

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

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

while(num)
{
num=num/10;
count++;
}

printf("Total digits is: %d",count);
return 0;
}
```

Sample output:
Enter a number: 23
Total digits is: 2

Write a c program to swap two numbers without using third variable.

```
#include<stdio.h>

int main()
{

int a=5,b=10;

//Solution one
a=b+a;
b=a-b;
a=a-b;
printf("a= %d b= %d",a,b);

// Solution two
a=5;b=10;
a=a^b;
b=a^b;
a=b^a;
printf("\na= %d b= %d",a,b);

return 0;
}
```

Write a C program which convert decimal number into Octal,Binary and Hexadecimal number.

```
#include<stdio.h>

int hex(int ino)
{
int imod=0;

if(ino)
{
imod=ino%16;
ino=ino/16;
hex(ino);
}

//print the mod and if mod is >9 then use letters A to F

if(imod>9)
{
```

```
switch(imod)
{
case 10:printf("A");break;
case 11:printf("B");break;
case 12:printf("C");break;
case 13:printf("D");break;
case 14:printf("E");break;
case 15:printf("F");break;
}
}
else
{
printf("%d",imod);
}
return 0;
}

int binary(int ino)
{
int imod=0;

if(ino)
{
imod=ino%2;
ino=ino/2;
binary(ino);
}

printf("%d",imod);
return 0;
}

int octal(int ino)
{

int imod=0;
if(ino)
{
imod=ino%8;
ino=ino/8;
octal(ino);
```

```
}
printf("%d",imod);
return 0;
}

int main(int argc, char* argv[])
{
int ino=0;

printf("\nPlease Enter the number");
scanf("%d",&ino);

printf("Number conversion.....\n");
printf("\nBinary format is\t");
binary(ino);

printf("\nOctal format is \t");
octal(ino);

printf("\nHexadecimal format is\t");
hex(ino);

return 0;
}
```

Write a c program to convert the string from upper case to lower case.

```
#include<stdio.h>
#include<string.h>
int main()
{
char str[20];
int i;
printf("Enter any string->");
scanf("%s",str);

printf("The string is->%s",str);

for(i=0;i<=strlen(str);i++)
{
if(str[i]>=65&&str[i]<=90)
```

```
{
str[i]=str[i]+32;
}

}

printf("\nThe string in lower case is->%s",str);
return 0;
}
```

Write a c program to convert the string from lower case to upper case.

```
#include<stdio.h>
int main(){
char str[20];
int i;
printf("Enter any string->");
scanf("%s",str);

printf("The string is->%s",str);

for(i=0;i<=strlen(str);i++)
{
if(str[i]>=97&&str[i]<=122)
{
str[i]=str[i]-32;
}
}

printf("\nThe string in lowercase is->%s",str);

return 0;
}
```

Write a C program which toggle the case of a string.

```
#include<stdio.h>
int main(){
char str[20];
int i;
printf("Enter any string->");
scanf("%s",str);
```

```
printf("The string is->%s",str);

for(i=0;i<=strlen(str);i++)
{
if(str[i]>=97&&str[i]<=122)
{
str[i]=str[i]-32;
}
else if(str[i]>=65&&str[i]<=90)
{
str[i]=str[i]+32;
}
}

printf("\nThe string in toggled case is->%s",str);

return 0;
}
```

Write a c program to find the size of int,char,float & double without using sizeof operator.

```
#include<stdio.h>

int main()
{

char *cptr = 0;
int *iptr = 0;
float *fptr = 0;
double *dptr = 0;

cptr++;
iptr++;
fptr++;
dptr++;

printf("Size of char data type: %d",cptr);
printf("Size of int data type: %d",iptr);
printf("Size of float data type: %d",fptr);
printf("Size of double data type: %d",dptr);
```

```
return 0;

}
```

Write a c program to find the size of structure & union without using sizeof operator

```
#include<stdio.h>

struct struct_demo{
int no;
float fno;
char c[20];
};

union union_demo{
int no;
float fno;
char c[20];
};

int main()
{

struct struct_demo *sptr = 0;
union union_demo *uptr = 0;

sptr++;
uptr++;

printf("Size of the structure struct_demo: %d",sptr);
printf("Size of the union union_demo: %d\n",uptr);

return 0;
}
```

Write a C program which accept password from the user.

```
#include<stdio.h>

int main()
{
```

```
char password[30];
char p;
int i=0;

printf("Enter the password:");

while((p=getch())!= 13)
{
password[i++] = p;
printf("*");
}

password[i] ='\0';

if(i<6)
{
printf("\nWeak password");
}

printf("\nYou have entered: %s",password);

return 0;
}
```

Sample output:

Enter the password: **********

You have entered: preplacement

Write a c program to open a file and write some text and close its by using library functions.

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

FILE *fp;
char ch;
```

```
// Open the file in write mode.
fp = fopen("file.txt","w");

printf("\nEnter data to be stored in to the file:");

while((ch=getchar())!=EOF)
{
putc(ch,fp);
}

fclose(fp);

return 0;
}
```

Write a c program to copy a data of file to other file by using library functions.

```
#include<stdio.h>
int main()
{
FILE *p,*q;

char file1[100],file2[100];
char ch;

// If file is not in the current directory then
// Provide the absolute path like D:\Home\source.txt
printf("\nEnter the source file name to be copied:");
// It accept the file name till we enter the enter key
gets(file1);

p=fopen(file1,"r");
if(p==NULL)
{
printf("cannot open %s",file1);
exit(0);
}

printf("\nEnter the destination file name:");
gets(file2);
```

```
q=fopen(file2,"w");
if(q==NULL)
{
printf("cannot open %s",file2);
exit(0);
}

//Copy the contents from source file to the
//destination file character by character
while((ch=getc(p))!=EOF)
{
putc(ch,q);
}

printf("\nCOMPLETED");

// Close the handle of opened files
fclose(p);
fclose(q);

return 0;
}
```

Write a c program which writes string in the file.

```
#include<stdio.h>
int main()
{
char str[70];
FILE *p;

// File opened in the read mode.
if((p=fopen("C:\demo.txt","r"))==NULL)
{
printf("\nUnable t open file string.txt");
exit(1);
}

// fgets() function read the contents from the file
// till it get the newline character.

// In our example we read the contents from a file whose
```

```
// file pointer is p and a\contents are copied into the
// character array str.

while(fgets(str,70,p)!=NULL)
{
puts(str);
}

fclose(p);

return 0;
}
```

Write a c program which writes array in the file.

```
#include<stdio.h>
int main()
{
FILE *p;

int i,a[10];

// Open the file in binary write mode
if((p=fopen("myfile.dat","wb"))==NULL)
{
printf("\nUnable to open file myfile.dat");
exit(1);
}

printf("\nEnter ten values, one value on each line\n");

// Accept the elemets of the array from the user
for(i=0;i<10;i++)
{
scanf("%d",&a[i]);
}

// Write the whole array inside the file
// Using same manner we can also write the
// structure and union inside the file
fwrite(a,sizeof(a),1,p);
```

```
fclose(p);
return 0;
}
```

Write a c program which writes array in the file.

```
#include <time.h>
#include <sys\stat.h>
#include <stdio.h>

void main()
{
struct stat status;

FILE *fp;

fp=fopen("test.txt","r");
fstat(fileno(fp),&status);

printf("Size of file : %d",status.st_size);
printf("Drive name : %c",65+status.st_dev);

}
```

Explanation:
Function int fstat (char *, struct stat *) store the information of open file in form of structure struct stat Structure struct stat has been defined in sys\stat.h as

```
struct stat
{
short st_dev, st_ino;
short st_mode, st_nlink;
int st_uid, st_gid;
short st_rdev;
long st_size, st_atime;
long st_mtime, st_ctime;
};
```

For more information read the book Advanced UNIX Programming By W. Richard Stevans

Write a C program to find the largest element in an

array.

```
#include<stdio.h>
int main()
{
int a[50],size,i,big;

printf("\nEnter the size of the array: ");
scanf("%d",&size);

printf("\nEnter %d elements in to the array: ", size);

// Accept the elemnts of a array from user
for(i=0;i<size;i++)
{
scanf("%d",&a[i]);
}

// Consider first element as a largest element
big=a[0];

for(i=1;i<size;i++)
{
if(big<a[i])
{
big=a[i];
}
}

printf("\nBiggest Element is : %d",big);

return 0;
}
```

Write a c program which deletes the duplicate element of an array.

```
#include<stdio.h>
int main()
{
int arr[50];
```

```
int *p;
int i,j,k,size,n;

printf("\nEnter size of the array: ");
scanf("%d",&n);

printf("\nEnter %d elements into the array: ",n);

// Accept the elements of the array from user
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}

// Copy the size of the array inside the size variable
size=n;

// Pointer p is now pointing at the first element of the array
// because array name is itself the base address of the array
p=arr;

// Use nested for loop becase we have to compaire each element
// of the array with all the elements of the array.
for(i=0;i<size;i++)
{
for(j=0;j<size;j++)
{
// If both the index are same means elment is also same then do nothing.
if(i==j)
{
continue;
}
// Compaire the elements and if they are equal(duplicate) do following things
else if(*(p+i)==*(p+j))
{
k=j;
// Decrement the size of array by 1
size--;
```

```
// Remove the duplicate element means shift the all the elements
// of the array by one position earlier
while(k < size)
{
*(p+k)=*(p+k+1);
k++;
}
j=0;
}
}
}

// Print the updated array after removing the duplicate elements.
printf("\nThe array after removing duplicates is: ");
for(i=0;i < size;i++)
{
printf(" %d",arr[i]);
}

return 0;
}
```

Write a C program to check whether our machine architecture is Little Endian or Big Endian.

```
#include<stdio.h>
int main()
{
int no = 51;
char *p;
p = &no;

if(*p == 0)
{
printf("Big endian");
}
else
{
printf("Little endian");
}
```

```
return 0;
}
```

Write a C program to check whether given strings are Anagram strings or not.

```
#include <stdio.h>

int check_anagram(char [], char []);

int main()
{
char first[100], second[100];
int flag;

printf("Enter first string\n");
gets(first);

printf("Enter second string\n");
gets(second);

flag = check_anagram(first, second);

if (flag == 1)
{
printf("\"%s\"first and \"%s\"are anagrams.\n",first, second);
}
else
{
printf("\"%s\"firstnd \"%s\"are not anagrams.\n",first,
second);
}

return 0;
}

int check_anagram(char first[], char second[])
{
int first_small[26] = {0}, second_small[26] = {0}, c = 0;
int first_capital[26] = {0}, second_capital[26] = {0};

// Traverse the first string and record the occurrences of each
and every character
```

```
while (first[c] != '\0')
{
// If the character in the first string is lower case
if(first[c]>= 'a' && first[c]<= 'z')
{
first_small[first[c]-'a']++;
}
// If the character in the first string is upper case
else if(first[c]>= 'A' && first[c]<= 'Z')
{
first_capital[first[c]-'A']++;
}
c++;
}

// Reinitialize the counter
c = 0;

// Traverse the second string and record the occurrences of each and every character
while (second[c] != '\0')
{
// If the character in the second string is lower case
if(second[c]>= 'a' && second[c]<= 'z')
{
second_small[second[c]-'a']++;
}
// If the character in the second string is upper case
else if(second[c]>= 'A' && second[c]<= 'Z')
{
second_capital[second[c]-'A']++;
}
c++;
}

// Traverse a loop for checking all the alphabets.
for (c = 0; c < 26; c++)
{
// If number of occurrences of particular character is not matched
if ((first_small[c] != second_small[c]) ||(first_capital[c] !=
```

```
second_capital[c]) )
{
return 0;
}
}

return 1;
}
```

Anagram String :

Two words are said to be anagrams of each other if the letters from one word can be rearranged to form the other word. From the above definition it is clear that two strings are anagrams if all characters in both strings occur same number of times. For example "abc" and "cab" are anagram strings, here every character 'a', 'b' and 'c' occur only one time in both strings. (In the above code case sensitivity is provided)

Fibonacci Series:

```
#include <stdio.h>
int main()
{
int i, n, t1 = 0, t2 = 1, nextTerm;


printf(“Enter the number of terms: “);
scanf(“%d”, &n);


printf(“Fibonacci Series: “);


for (i = 1; i <= n; ++i)
{
   printf(“%d, “, t1);
   nextTerm = t1 + t2;
   t1 = t2;
```

```
    t2 = nextTerm;
}
    return 0;
}
```

Output

```
Enter the number of terms: 10
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

C Program to add two numbers without using arithmetic operator

```
#include<stdio.h>

int Add(int x, int y)
{
    // Iterate till there is no carry
    while (y != 0)
    {
        // carry now contains common
        //set bits of x and y
        int carry = x & y;

        // Sum of bits of x and y where at
        //least one of the bits is not set
        x = x ^ y;

        // Carry is shifted by one so that adding
        // it to x gives the required sum
        y = carry << 1;
    }
    return x;
}

int main()
{
    printf("%d", Add(10, 32));
    return 0;
}
```

Output :

42