

Some Important C Programs

Prepared by : MD. SHAHIDUL ISLAM (PAPPU)
KHULNA UNIVERCITY
C.S.E
ID:140225

Commented [MS11]:

Program to convert decimal to binary in c

Decimal to binary conversion in c programming language. C source code for decimal to binary conversion:

```
#include<stdio.h>

int main(){
    long int decimalNumber,remainder,quotient;
    int binaryNumber[100],i=1,j;

    printf("Enter any decimal number: ");
    scanf("%ld",&decimalNumber);

    quotient = decimalNumber;

    while(quotient!=0){
        binaryNumber[i++]= quotient % 2;
        quotient = quotient / 2;
    }

    printf("Equivalent binary value of decimal number %d: ",decimalNumber);
```

```
for(j = i -1 ;j> 0;j--)\n\n    printf("%d",binaryNumber[j]);\n\nreturn 0;\n}\n
```

Sample output:

Enter any decimal number: 50
Equivalent binary value of decimal number 50: 110010

Algorithm:

Binary number system: It is base 2 number system which uses the digits from 0 and 1.

Decimal number system:

It is base 10 number system which uses the digits from 0 to 9

Convert from decimal to binary algorithm:

Following steps describe how to convert decimal to binary

Step 1: Divide the original decimal number by 2
Step 2: Divide the quotient by 2
Step 3: Repeat the step 2 until we get quotient equal to zero.

Equivalent binary number would be remainders of each step in the reverse order.

Decimal to binary conversion with example:

For example we want to convert decimal number 25 in the binary.

Step 1: 25 / 2 Remainder : 1 , Quotient : 12
Step 2: 12 / 2 Remainder : 0 , Quotient : 6
Step 3: 6 / 2 Remainder : 0 , Quotient : 3
Step 4: 3 / 2 Remainder : 1 , Quotient : 1
Step 5: 1 / 2 Remainder : 1 , Quotient : 0

So equivalent binary number is: 11001
That is $(25)_{10} = (11001)_2$

C code for decimal to octal converter

```
#include<stdio.h>

int main(){

    long int decimalNumber,remainder,quotient;
    int octalNumber[100],i=1,j;

    printf("Enter any decimal number: ");
    scanf("%ld",&decimalNumber);

    quotient = decimalNumber;

    while(quotient!=0){
        octalNumber[i++]= quotient % 8;
        quotient = quotient / 8;
    }

    printf("Equivalent octal value of decimal number %d: ",decimalNumber);
    for(j = i -1 ;j> 0;j--)
        printf("%d",octalNumber[j]);

    return 0;
}
```

Sample output:

```
Enter any decimal number: 50
Equivalent octal value of decimal number 50: 62
```

2. Easy way to convert decimal number to octal number in c

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

    long int decimalNumber;

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

    printf("Equivalent octal number is: %o",decimalNumber);
}
```

```
    return 0;
}
```

Sample output:

Enter any decimal number: 25
Equivalent octal number is: 31

Octal number system: It is base 8 number system which uses the digits from 0 to 7.

Decimal number system:
It is base 10 number system which uses the digits from 0 to 9

Decimal to octal conversion method:

Step 1: Divide the original decimal number by 8
Step 2: Divide the quotient by 8
Step 3: Repeat the step 2 until we get quotient equal to zero.

Result octal number would be remainders of each step in the reverse order.

Decimal to octal conversion with example:

For example we want to convert decimal number 525 in the octal.

Step 1: 525 / 8 Remainder : 5 , Quotient : 65
Step 2: 65 / 8 Remainder : 1 , Quotient : 8
Step 3: 8 / 8 Remainder : 0 , Quotient : 1
Step 4: 1 / 8 Remainder : 1 , Quotient : 0

So equivalent octal number is: 1015
That is $(525)_{10} = (1015)_8$

1. C code to convert decimal to hexadecimal

```
#include<stdio.h>
int main(){
    long int decimalNumber,remainder,quotient;
    int i=1,j,temp;
    char hexadecimalNumber[100];

    printf("Enter any decimal number: ");
```

```

scanf("%ld",&decimalNumber);

quotient = decimalNumber;

while(quotient!=0){
    temp = quotient % 16;

    //To convert integer into character
    if( temp < 10)
        temp =temp + 48;
    else
        temp = temp + 55;

    hexadecimalNumber[i++] = temp;
    quotient = quotient / 16;
}

printf("Equivalent hexadecimal value of decimal number %d: ",decimalNumber);
for(j = i -1 ;j> 0;j--){
    printf("%c",hexadecimalNumber[j]);
}

return 0;
}

```

Sample output:

```

Enter any decimal number: 45
Equivalent hexadecimal value of decimal number 45: 2D

```

2. Easy way to convert decimal number to hexadecimal number:

```

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

    long int decimalNumber;

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

    printf("Equivalent hexadecimal number is: %X",decimalNumber);

    return 0;
}

```

Sample output:

Enter any decimal number: 45
Equivalent hexadecimal number is: 2D

Hexadecimal number system: It is base 16 number system which uses the digits from 0 to 9 and A, B, C, D, E, F.

Decimal number system:

It is base 10 number system which uses the digits from 0 to 9

Decimal to hexadecimal conversion method:

Following steps describe how to convert decimal to hexadecimal

- Step 1: Divide the original decimal number by 16
- Step 2: Divide the quotient by 16
- Step 3: Repeat the step 2 until we get quotient equal to zero.

Equivalent binary number would be remainders of each step in the reverse order.

Decimal to hexadecimal conversion example:

For example we want to convert decimal number 900 in the hexadecimal.

Step 1: 900 / 16 Remainder : 4 , Quotient : 56
Step 2: 56 / 16 Remainder : 8 , Quotient : 3
Step 3: 3 / 16 Remainder : 3 , Quotient : 0

So equivalent hexadecimal number is: 384
That is $(900)_{10} = (384)_{16}$

C program to convert binary to octal

```
#include<stdio.h>
#define MAX 1000
int main(){
```

```

char octalNumber[MAX];
long int i=0;

printf("Enter any octal number: ");
scanf("%s",octalNumber);

printf("Equivalent binary value: ");
while(octalNumber[i]){
    switch(octalNumber[i]){
        case '0': printf("000"); break;
        case '1': printf("001"); break;
        case '2': printf("010"); break;
        case '3': printf("011"); break;
        case '4': printf("100"); break;
        case '5': printf("101"); break;
        case '6': printf("110"); break;
        case '7': printf("111"); break;
        default: printf("\nInvalid octal digit %c",octalNumber[i]); return 0;
    }
    i++;
}

return 0;
}

```

Sample output:

```

Enter any octal number: 123
Equivalent binary value: 001010011

```

Algorithm:

Octal to binary conversion method:

To convert or change the octal number to binary number replace the each octal digits by a binary number using octal to binary chart.

| Octal | Binary |
|-------|--------|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

Octal to binary table

Octal to binary conversion examples:

For example we want to convert or change octal number 65201 to decimal.
For this we will replace each octal digit to binary values using the above table:

Octal number: 6 5 2 0 1
Binary values: 110 101 010 000 001

So $(65201)_8 = (110101010000001)_2$

C program for hexadecimal to binary conversion

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

int main(){
    char binaryNumber[MAX],hexaDecimal[MAX];
    long int i=0;

    printf("Enter any hexadecimal number: ");
    scanf("%s",hexaDecimal);

    printf("\nEquivalent binary value: ");
    while(hexaDecimal[i]){
        switch(hexaDecimal[i]){
            case '0': printf("0000"); break;
            case '1': printf("0001"); break;
            case '2': printf("0010"); break;
            case '3': printf("0011"); break;
            case '4': printf("0100"); break;
            case '5': printf("0101"); break;
            case '6': printf("0110"); break;
            case '7': printf("0111"); break;
            case '8': printf("1000"); break;
            case '9': printf("1001"); break;
            case 'A': printf("1010"); break;
            case 'B': printf("1011"); break;
            case 'C': printf("1100"); break;
            case 'D': printf("1101"); break;
            case 'E': printf("1110"); break;
            case 'F': printf("1111"); break;
        }
    }
}
```



```

        case 'a': printf("1010"); break;
        case 'b': printf("1011"); break;
        case 'c': printf("1100"); break;
        case 'd': printf("1101"); break;
        case 'e': printf("1110"); break;
        case 'f': printf("1111"); break;
        default: printf("\nInvalid hexadecimal digit %c",hexadecimal[i]); return 0;
    }
    i++;
}

return 0;
}

```

Sample output:

Enter any hexadecimal number: 2AD5
 Equivalent binary value: 0010101011010101

Algorithm:

Hexadecimal to binary conversion method:

To convert or change the hexadecimal number to binary number replace the each octal digits by a binary number using hexadecimal to binary chart.

| Hexadecimal | Binary |
|-------------|--------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

Hexadecimal to binary table

Hexadecimal to binary conversion examples:

For example we want to convert or change hexadecimal number 65B2 to binary. For this we will replace each hexadecimal digit to binary values using the above table:

| | | | | |
|---------------------|------|------|------|------|
| Hexadecimal number: | 6 | 5 | B | 2 |
| Binary values: | 0110 | 0101 | 1011 | 0010 |

So $(65B2)_{16} = (0110010110110010)_2$

C program to convert binary to octal

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

    long int binaryNumber,octalNumber=0,j=1,remainder;

    printf("Enter any number any binary number: ");
    scanf("%ld",&binaryNumber);

    while(binaryNumber!=0){
        remainder=binaryNumber%10;
        octalNumber=octalNumber+remainder*j;
        j=j*2;
        binaryNumber=binaryNumber/10;
    }

    printf("Equivalent octal value: %lo",octalNumber);

    return 0;
}
```

Sample output:

```
Enter any number any binary number: 1101
Equivalent octal value: 15
```

C code for how to convert large binary to octal

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

int main(){

    char binaryNumber[MAX],octalNumber[MAX];
    long int i=0,j=0;
```

[illegible][illegible]

Step 2: Replace the each group with following values:

| Binary number | Octal values |
|---------------|--------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Binary to octal chart

Binary to octal conversion examples:

For example we want to convert binary number 1011010101001101 to octal.

Step 1: 001 011 010 101 001 101

Step 2: 1 3 2 5 1 5

So $(1011010101001101)_2 = (132515)_8$

C code to convert octal number to decimal number

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

int main(){

    long int octal,decimal =0;
    int i=0;

    printf("Enter any octal number: ");
    scanf("%ld",&octal);

    while(octal!=0){
        decimal = decimal + (octal % 10) * pow(8,i++);
        octal = octal/10;
    }

    printf("Equivalent decimal value: %ld",decimal);

    return 0;
}
```

Sample output:

Enter any octal number: 346
Equivalent decimal value: 230

C program to change octal to decimal

```
#include<stdio.h>

int main(){

    long int octalNumber;

    printf("Enter any octal number: ");
    scanf("%o",&octalNumber);

    printf("Equivalent decimal number is: %d",octalNumber);

    return 0;
}
```

Sample output:

Enter any octal number: 17
Equivalent decimal number is: 15

Algorithm:

Octal to decimal conversion method:

To convert an octal number to decimal number multiply each digits separately of octal number from right side by $8^0, 8^1, 8^2, 8^3 \dots$ respectively and then find out the sum of them.

Octal to decimal conversion examples:

For example we want to convert octal number 3401 to decimal number.

Step 1: $1 * 8^0 = 1 * 1 = 1$

Step 2: $0 * 8^1 = 0 * 8 = 0$

Step 3: $4 * 8^2 = 4 * 64 = 256$

Step 4: $3 * 8^3 = 3 * 512 = 1536$

Sum= $1 + 0 + 256 + 1536 = 1793$

So, $(3401)_8 = (1793)_{10}$

Write a c program to convert binary number to decimal number

C code for binary to decimal conversion:

```
#include<stdio.h>

int main(){

    long int binaryNumber,decimalNumber=0,j=1,remainder;

    printf("Enter any number any binary number: ");
    scanf("%ld",&binaryNumber);

    while(binaryNumber!=0){
        remainder=binaryNumber%10;
        decimalNumber=decimalNumber+remainder*j;
        j=j*2;
        binaryNumber=binaryNumber/10;
    }

    printf("Equivalent decimal value: %ld",decimalNumber);

    return 0;
}
```

Sample output:

```
Enter any number any binary number: 1101
Equivalent decimal value: 13
```

Algorithm:

Binary number system: It is base 2 number system which uses the digits from 0 and 1.

Decimal number system:

It is base 10 number system which uses the digits from 0 to 9

Convert from binary to decimal algorithm:

For this we multiply each digit separately from right side by 1, 2, 4, 8, 16 ... respectively then add them.

Binary number to decimal conversion with example:

For example we want to convert binary number 101111 to decimal:

```
Step1:  1 * 1 = 1
Step2:  1 * 2 = 2
Step3:  1 * 4 = 4
Step4:  1 * 8 = 8
Step5:  0 * 16 = 0
Step6:  1 * 32 = 32
```

Its decimal value: $1 + 2 + 4 + 8 + 0 + 32 = 47$

That is $(101111)_2 = (47)_{10}$

Binary to hexadecimal conversion in c

C program for hexadecimal to binary conversion

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

    long int binaryNumber,hexadecimalNumber=0,j=1,remainder;

    printf("Enter any number any binary number: ");
    scanf("%ld",&binaryNumber);

    while(binaryNumber!=0){
        remainder=binaryNumber%10;
        hexadecimalNumber=hexadecimalNumber+remainder*j;
        j=j*2;
        binaryNumber=binaryNumber/10;
    }

    printf("Equivalent hexadecimal value: %lX",hexadecimalNumber);

    return 0;
}
```

Sample output:

```
Enter any number any binary number: 1101
Equivalent hexadecimal value: D
```

How to convert large binary number to hexadecimal

```

#include<stdio.h>
#define MAX 1000

int main(){

    char binaryNumber[MAX],hexadecimal[MAX];
    int temp;
    long int i=0,j=0;

    printf("Enter any number any binary number: ");
    scanf("%s",binaryNumber);

    while(binaryNumber[i]){
        binaryNumber[i] = binaryNumber[i] -48;
        ++i;
    }

    --i;
    while(i-2>=0){
        temp = binaryNumber[i-3] *8 + binaryNumber[i-2] *4
+ binaryNumber[i-1] *2 + binaryNumber[i] ;
        if(temp > 9)
            hexadecimal[j++] = temp + 55;
        else
            hexadecimal[j++] = temp + 48;
        i=i-4;
    }

    if(i ==1)
        hexadecimal[j] = binaryNumber[i-1] *2 + binaryNumber[i] + 48 ;
    else if(i==0)
        hexadecimal[j] = binaryNumber[i] + 48 ;
    else
        --j;

    printf("Equivalent hexadecimal value: ");
    while(j>=0){
        printf("%c",hexadecimal[j--]);
    }

    return 0;
}

```

Sample output:

```

Enter any number any binary number: 1010011011100011110
001001111011110001000100011101110111011110
Equivalent hexadecimal value: 14DC789EF111DDDE

```

Algorithm:

Binary to hexadecimal conversion method

Step1: Arrange the binary number in the group 4 from right side.

Step 2: Replace the each group with following values:

| Binary number | Hexadecimal values |
|---------------|--------------------|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | A |
| 1011 | B |
| 1100 | C |
| 1101 | D |
| 1110 | E |
| 1111 | F |

Binary to hexadecimal chart

Binary to hexadecimal conversion examples:

For example we want to convert binary number 11011010101001101 to hexadecimal.

Step 1: 0001 1011 0101 0100 1101
Step 2: 1 B 5 4 D

So $(11011010101001101)_2 = (1B54D)_{16}$

C program for addition of binary numbers

C code for sum of two binary numbers:

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

```

int main(){

    long int binary1,binary2;
    int i=0,remainder = 0,sum[20];

    printf("Enter any first binary number: ");
    scanf("%ld",&binary1);
    printf("Enter any second binary number: ");
    scanf("%ld",&binary2);

    while(binary1!=0||binary2!=0){
        sum[i++] = (binary1 %10 + binary2 %10 + remainder ) % 2;
        remainder = (binary1 %10 + binary2 %10 + remainder ) / 2;
        binary1 = binary1/10;
        binary2 = binary2/10;
    }

    if(remainder!=0)
        sum[i++] = remainder;

    --i;
    printf("Sum of two binary numbers: ");
    while(i>=0)
        printf("%d",sum[i--]);

    return 0;
}

```

Sample output:

```

Enter any first binary number: 1100011
Enter any second binary number: 1101
Sum of two binary numbers: 1110000

```

Alogrithm:

Rule of binary addition:

```

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 1 and carry = 1

```

Q1. What is the sum of the binary numbers 1101 and 1110?

Answer: 1101 + 1110 = 11011

C program for multiplication of two binary numbers

C code for product of two binary numbers

C program for multiplication of two binary numbers

```
#include<stdio.h>

int binaryAddition(int,int);

int main(){

    long int binary1,binary2,multiply=0;
    int digit,factor=1;

    printf("Enter any first binary number: ");
    scanf("%ld",&binary1);
    printf("Enter any second binary number: ");
    scanf("%ld",&binary2);

    while(binary2!=0){
        digit =  binary2 %10;

        if(digit ==1){
            binary1=binary1*factor;
            multiply = binaryAddition(binary1,multiply);
        }
        else
            binary1=binary1*factor;

        binary2 = binary2/10;
        factor = 10;
    }

    printf("Product of two binary numbers: %ld",multiply);

    return 0;
}

int binaryAddition(int binary1,int binary2){

    int i=0,remainder = 0,sum[20];
    int binarySum=0;

    while(binary1!=0||binary2!=0){
        sum[i++] =  (binary1 %10 + binary2 %10 + remainder ) % 2;
        remainder = (binary1 %10 + binary2 %10 + remainder ) / 2;
```

```

        binary1 = binary1/10;
        binary2 = binary2/10;
    }

    if(remainder!=0)
        sum[i++] = remainder;
    --i;
    while(i>=0)
        binarySum = binarySum*10 + sum[i--];

    return binarySum;
}

```

Sample output:

```

Enter any first binary number: 1101
Enter any second binary number: 11
Product of two binary numbers: 100111

```

Algorithm:

Rule of binary multiplication:

```

0 * 0 = 0
1 * 0 = 0
0 * 1 = 0
1 * 1 = 1

```

Q. what is the product of the binary numbers 0110 and 0011?

Answer: 0110 * 0011 = 10010

C program fractional binary conversion from decimal

C code for fractional binary to decimal converter:

```

#include<stdio.h>
#define MAX 1000

int main(){

    long double fraDecimal=0.0,dFractional=0.0 ,fraFactor=0.5;
    long int dIntegral = 0,bIntegral=0,bFractional[MAX];
    long int intFactor=1,remainder,i=0,k=0,flag=0;

```

```

char fraBinary[MAX];

printf("Enter any fractional binary number: ");
scanf("%s",&fraBinary);

while(fraBinary[i]){

    if(fraBinary[i] == '.')
        flag = 1;
    else if(flag==0)
        bIntegral = bIntegral * 10 + (fraBinary[i] -48);
    else
        bFractional[k++] = fraBinary[i] -48;
    i++;
}

while(bIntegral!=0){
    remainder=bIntegral%10;
    dIntegral= dIntegral+remainder*intFactor;
    intFactor=intFactor*2;
    bIntegral=bIntegral/10;
}

for(i=0;i<k;i++){
    dFractional = dFractional + bFractional[i] * fraFactor;
    fraFactor = fraFactor / 2;
}

fraDecimal = dIntegral + dFractional ;

printf("Equivalent decimal value: %Lf",fraDecimal);

return 0;
}

```

Sample output:

```

Enter any fractional binary number: 11.11
Equivalent decimal value: 3.750000

```

Algorithm:

Algorithm to convert the fractional binary to decimal or floating point binary to decimal:

Step1. First we convert the integral part of binary number to decimal. For this we multiply each digit separately from right side by 1, 2, 4, 8, 16 ... respectively then add them this is integral part of decimal number.

Step2. Now we convert the fractional part of binary number to decimal. For this we multiply each digit separately from left side by $1/2$, $1/4$, $1/8$, $1/16$... respectively then add them this is fractional part of decimal number.

Step3: Add the integral and fractional part of decimal number.

Example for floating point binary to decimal:

For example we want to convert the binary number 101111.1101 to decimal number.

Step1: Conversions of 101111 to decimal:

```
S1: 1 * 1 = 1
S2: 1 * 2 = 2
S3: 1 * 4 = 4
S4: 1 * 8 = 8
S5: 0 * 16 = 0
S6: 1 * 32 = 32
```

Integral part of decimal number: $1 + 2 + 4 + 8 + 0 + 32 = 47$

Step2: Conversions of .1101 to decimal:

```
S1: 1 * (1/2) = 0.5
S2: 1 * (1/4) = 0.25
S3: 0 * (1/8) = 0
S4: 1 * (1/16) = 0.0625
```

Fractional part of decimal number = $0.5 + 0.25 + 0 + 0.0625 = 0.8125$

So equivalent binary number is: 0.101100

Step 3: So binary value of binary number 101111.1101 will be $47 + 0.8125 = 47.8125$

C program for fractional decimal to binary fraction conversion

C code for fractional decimal to binary converter:

```

#include<stdio.h>

int main(){
    long    double    fraDecimal,fraBinary,bFractional    =
0.0,dFractional,fraFactor=0.1;
    long int dIntegral,bIntegral=0;
    long int intFactor=1,remainder,temp,i;

    printf("Enter any fractional decimal number: ");
    scanf("%Lf",&fraDecimal);

    dIntegral = fraDecimal;
    dFractional = fraDecimal - dIntegral;

    while(dIntegral!=0){
        remainder=dIntegral%2;
        bIntegral=bIntegral+remainder*intFactor;
        dIntegral=dIntegral/2;
        intFactor=intFactor*10;
    }

    for(i=1;i<=6;i++){
        dFractional = dFractional * 2;
        temp = dFractional;

        bFractional = bFractional + fraFactor* temp;
        if(temp ==1)
            dFractional = dFractional - temp;

        fraFactor=fraFactor/10;
    }

    fraBinary = bIntegral + bFractional;
    printf("Equivalent binary value: %lf",fraBinary);

    return 0;
}

```

Sample output:

```

Enter any fractional decimal number: 5.7
Equivalent binary value: 101.101100

```

Algorithm:

How to convert fractional decimal to binary:

Following steps describe how to convert decimal to binary

Step1. First we convert the integral part of binary number to decimal.

Following steps describe how to convert decimal number to binary number:

S1: Divide the decimal number by 2
S2: Divide the quotient by 2
S3: Repeat the step 2 until we get quotient equal to zero.

Equivalent binary number would be remainders of each step in the reverse order.

Step2. Now we convert the fractional part of decimal number to binary.
Following steps describe how to convert floating decimal to binary

S1: Multiply the decimal number by 2
S2: Integral part of resultant decimal number will be first digit of fraction binary number.
S3: Repeat the S1 using only fractional part of decimal number and then S2.

Step3: Add the integral and fractional part of binary number.

Example for floating point decimal to binary:

For example we want to convert the decimal number 25.7 to binary number.

Step1: Conversions of 25 to binary.

S1: 25 / 2 Remainder : 1 , Quotient : 12
S2: 12 / 2 Remainder : 0 , Quotient : 6
S3: 6 / 2 Remainder : 0 , Quotient : 3
S4: 3 / 2 Remainder : 1 , Quotient : 1
S5: 1 / 2 Remainder : 1 , Quotient : 0

So equivalent binary number is: 11001

Step2: Conversions of .7 to binary.

S1: $0.7 * 2 = 1.4$, Integral part = 1
S2: $0.4 * 2 = 0.8$, Integral part = 0
S3: $0.8 * 2 = 1.6$, Integral part = 1
S4: $0.6 * 2 = 1.2$, Integral part = 1
S5: $0.2 * 2 = 0.4$, Integral part = 0
S6: $0.4 * 2 = 0.8$, Integral part = 0

So equivalent binary number is: 0.101100

Step 3: So binary value of decimal number 25.7 will be
 $11001 + 0.101100 = 1101.101100$

C program to convert decimal to roman number

Convert numbers to roman numerals in c

```
#include<stdio.h>

void predigits(char c1,char c2);
void postdigits(char c,int n);

char roman_Number[1000];
int i=0;

int main(){

    int j;
    long int number;

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

    if(number <= 0){
        printf("Invalid number");
        return 0;
    }

    while(number != 0){

        if(number >= 1000){
            postdigits('M',number/1000);
            number = number - (number/1000) * 1000;
        }
        else if(number >=500){
            if(number < (500 + 4 * 100)){
                postdigits('D',number/500);
                number = number - (number/500) * 500;
            }
            else{
                predigits('C','M');
                number = number - (1000-100);
            }
        }
        else if(number >=100){
            if(number < (100 + 3 * 100)){
                postdigits('C',number/100);
                number = number - (number/100) * 100;
            }
        }
    }
}
```

```

    }
    else{
        predigits('L','D');
        number = number - (500-100);
    }
}
else if(number >=50){
    if(number < (50 + 4 * 10)){
        postdigits('L',number/50);
        number = number - (number/50) * 50;
    }
    else{
        predigits('X','C');
        number = number - (100-10);
    }
}
else if(number >=10){
    if(number < (10 + 3 * 10)){
        postdigits('X',number/10);
        number = number - (number/10) * 10;
    }
    else{
        predigits('X','L');
        number = number - (50-10);
    }
}
else if(number >=5){
    if(number < (5 + 4 * 1)){
        postdigits('V',number/5);
        number = number - (number/5) * 5;
    }
    else{
        predigits('I','X');
        number = number - (10-1);
    }
}
else if(number >=1){
    if(number < 4){
        postdigits('I',number/1);
        number = number - (number/1) * 1;
    }
    else{
        predigits('I','V');
        number = number - (5-1);
    }
}
}

printf("Roman number will be: ");
for(j=0;j<i;j++)

```

```

        printf("%c",roman_Number[j]);

    return 0;

}

void predigits(char c1,char c2){
    roman_Number[i++] = c1;
    roman_Number[i++] = c2;
}

void postdigits(char c,int n){
    int j;
    for(j=0;j<n;j++){
        roman_Number[i++] = c;
    }
}

```

Sample output:

Enter any natural number: 87
 Roman number will be: LXXXVII

C program to convert Roman number to decimal number

C code for roman numbers to English numbers

```

#include<stdio.h>
#include<string.h>

int digitValue(char);

int main(){

    char roman_Number[1000];
    int i=0;
    long int number =0;

    printf("Enter any roman number (Valid digits are I, V, X, L, C, D, M): \n");
    scanf("%s",roman_Number);

    while(roman_Number[i]){

        if(digitValue(roman_Number[i]) < 0){

```

```

        printf("Invalid roman digit : %c",roman_Number[i]);
        return 0;
    }

    if((strlen(roman_Number) -i) > 2){
        if(digitValue(roman_Number[i]) <
digitValue(roman_Number[i+2])){
            printf("Invalid roman number");
            return 0;
        }
    }

    if(digitValue(roman_Number[i]) >= digitValue(roman_Number[i+1]))
        number = number + digitValue(roman_Number[i]);
    else{
        number = number + (digitValue(roman_Number[i+1]) -
digitValue(roman_Number[i]));
        i++;
    }
    i++;
}

printf("Its decimal value is : %ld",number);

return 0;
}

int digitValue(char c){
    int value=0;

    switch(c){
        case 'I': value = 1; break;
        case 'V': value = 5; break;
        case 'X': value = 10; break;
        case 'L': value = 50; break;
        case 'C': value = 100; break;
        case 'D': value = 500; break;
        case 'M': value = 1000; break;
        case '\0': value = 0; break;
        default: value = -1;
    }

    return value;
}

```

Sample output:

Enter any roman number (Valid digits are I, V, X, L, C, D, M):

XVII

Its decimal value is: 17

C program to convert digits to words

C code to covert each digits of a number in English word
Convert digits to words in c

```
#include<stdio.h>

int main(){

    int number,i=0,j,digit;
    char * word[1000];

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

    while(number){

        digit = number %10;
        number = number /10;

        switch(digit){
            case 0: word[i++] = "zero"; break;
            case 1: word[i++] = "one"; break;
            case 2: word[i++] = "two"; break;
            case 3: word[i++] = "three"; break;
            case 4: word[i++] = "four"; break;
            case 5: word[i++] = "five"; break;
            case 6: word[i++] = "six"; break;
            case 7: word[i++] = "seven"; break;
            case 8: word[i++] = "eight"; break;
            case 9: word[i++] = "nine"; break;
        }

    }

    for(j=i-1;j>=0;j--){
        printf("%s ",word[j]);
    }

    return 0;
}
```

```
}
```

Sample output:

```
Enter any integer: 23451208
two three four five one two zero eight
```

C program to convert currency into words

C code to convert any number to English word:

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

void toWord(int,int);
char * getPositionValue(int);
char * digitToWord(int);

char word[100][30];
int i =0;

int main(){

    int j,k,subnumber;
    unsigned long int number;

    printf("Enter any postive number: ");
    scanf("%lu",&number);

    if(number ==0){
        printf("Zero");
        return 0;
    }

    while(number){

        if(i==1){
            toWord(number %10,i);
            number = number/10;
        }
    }
}
```

```

        }else{
            toWord(number %100,i);
            number = number/100;
        }

        i++;
    }

    printf("Number in word: ");
    *word[i-1] = *word[i-1] - 32;
    for(j=i-1;j>=0;j--){
        printf("%s",word[j]);
    }

    return 0;
}

void toWord(int number,int position){

    char numberToword[100]={" "};

    if(number ==0){
    }else if (number < 20 ||number %10==0){
        strcpy(numberToword,digitToWord(number));
    }else{
        strcpy(numberToword,digitToWord((number/10)*10));
        strcat(numberToword,digitToWord(number%10));
    }

    strcat(numberToword,getPositionValue(position));
    strcpy(word[i],numberToword);
}

char * getPositionValue(int postion){

    static char positionValue[10]=" ";

    switch(postion){

        case 1: strcpy(positionValue,"hundreds "); break;
        case 2: strcpy(positionValue,"thousand "); break;
        case 3: strcpy(positionValue,"lakh "); break;
        case 4: strcpy(positionValue,"crore "); break;
        case 5: strcpy(positionValue,"arab "); break;
        case 6: strcpy(positionValue,"kharab "); break;
        case 7: strcpy(positionValue,"neel "); break;
        case 8: strcpy(positionValue,"padam "); break;
    }
}

```

```

    return positionValue;
}

char * digitToWord(int digit){

    static char digitInWord[10]=" ";

    switch(digit){
        case 1: strcpy(digitInWord , "one "); break;
        case 2: strcpy(digitInWord , "two "); break;
        case 3: strcpy(digitInWord , "three "); break;
        case 4: strcpy(digitInWord , "four "); break;
        case 5: strcpy(digitInWord , "five "); break;
        case 6: strcpy(digitInWord , "six "); break;
        case 7: strcpy(digitInWord , "seven "); break;
        case 8: strcpy(digitInWord , "eight "); break;
        case 9: strcpy(digitInWord , "nine ");break;
        case 10: strcpy(digitInWord , "ten "); break;
        case 11: strcpy(digitInWord , "eleven "); break;
        case 12: strcpy(digitInWord , "twelve "); break;
        case 13: strcpy(digitInWord , "thirteen "); break;
        case 14: strcpy(digitInWord , "fourteen "); break;
        case 15: strcpy(digitInWord , "fifteen "); break;
        case 16: strcpy(digitInWord , "sixteen "); break;
        case 17: strcpy(digitInWord , "seventeen "); break;
        case 18: strcpy(digitInWord , "eighteen "); break;
        case 19: strcpy(digitInWord , "nineteen "); break;
        case 20: strcpy(digitInWord , "twenty "); break;
        case 30: strcpy(digitInWord , "thirty "); break;
        case 40: strcpy(digitInWord , "fourty "); break;
        case 50: strcpy(digitInWord , "fifty "); break;
        case 60: strcpy(digitInWord , "sixty "); break;
        case 70: strcpy(digitInWord , "seventy "); break;
        case 80: strcpy(digitInWord , "eighty "); break;
        case 90: strcpy(digitInWord,"ninety "); break;
    }

    return digitInWord;
}

```

CONCATENATION OF TWO STRINGS USING C PROGRAM

Concatenation of two strings in c programming language


```

#include<stdio.h>
int main(){
    int i=0,j=0;
    char str1[20],str2[20];
    puts("Enter first string");
    gets(str1);
    puts("Enter second string");
    gets(str2);
    printf("Before concatenation the strings are\n");
    puts(str1);
    puts(str2);
    while(str1[i]!='\0'){
        i++;
    }
    while(str2[j]!='\0'){
        str1[i++]=str2[j++];
    }
    str1[i]='\0';
    printf("After concatenation the strings are\n");
    puts(str1);
    return 0;
}

```

Program to convert string into ASCII values in c programming language:

```

#include<stdio.h>
int main(){
    char str[100];
    int i=0;
    printf("Enter any string: ");
    scanf("%s",str);

    printf("ASCII values of each characters of given string: ");
    while(str[i])
        printf("%d ",str[i++]);

    return 0;
}

```

Sample Output:

Enter any string: cquestionbank.blogspot.com

ASCII values of each characters of given string: 99 113 117 101 115 116 105
111 110 98 97 110 107 46 98 108 111 103 115 112 111 116 46 99 111 109

C program for atm transactions

1. C code for atm transaction while currencies are 1000,500 and 100

- 2. ATM c program source code
- 3. C program for atm machine
- 4. C program for atm banking

```
#include<stdio.h>

int totalThousand =1000;
int totalFiveFundred =1000;
int totalOneHundred =1000;

int main(){

    unsigned long withdrawAmount;
    unsigned long totalMoney;

    int thousand=0,fiveHundred=0,oneHundred=0;

    printf("Enter the amount in multiple of 100: ");
    scanf("%lu",&withdrawAmount);

    if(withdrawAmount %100 != 0){
        printf("Invalid amount;");
        return 0;
    }

    totalMoney = totalThousand * 1000 + totalFiveFundred* 500
+ totalOneHundred*100;

    if(withdrawAmount > totalMoney){
        printf("Sorry,Insufficient money");
        return 0;
    }

    thousand = withdrawAmount / 1000;
    if(thousand > totalThousand)
        thousand = totalThousand;
```

```

withdrawAmount = withdrawAmount - thousand * 1000;

if (withdrawAmount > 0){
    fiveHundred = withdrawAmount / 500;
    if(fiveHundred > totalFiveFundred)
        fiveHundred = totalFiveFundred;
    withdrawAmount = withdrawAmount - fiveHundred * 500;
}

if (withdrawAmount > 0)
    oneHundred = withdrawAmount / 100;

printf("Total 1000 note: %d\n",thousand);
printf("Total 500 note: %d\n",fiveHundred);
printf("Total 100 note: %d\n",oneHundred);

return 0;
}

```

Sample output:

```

Enter the amount in multiple of 100: 7800
Total 1000 note: 7
Total 500 note: 1
Total 100 note: 3

```

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

Sum of prime numbers is: 732

BINARY SEARCH USING C PROGRAM

1. Write a simple code for binary search in c programming language
2. Wap a c program to search an element in an array using binary search

```

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

    int a[10],i,n,m,c=0,l,u,mid;

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

    printf("Enter the elements in ascending order: ");
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }

    printf("Enter the number to be search: ");
    scanf("%d",&m);

    l=0,u=n-1;
    while(l<=u){
        mid=(l+u)/2;
        if(m==a[mid]){
            c=1;
            break;
        }
        else if(m<a[mid]){
            u=mid-1;
        }
    }
}

```

```

        else
            l=mid+1;
    }
    if(c==0)
        printf("The number is not found.");
    else
        printf("The number is found.");

    return 0;
}

```

Sample output:

Enter the size of an array: 5

Enter the elements in ascending order: 4 7 8 11 21

Enter the number to be search: 11

The number is found.

C program for binary search using recursion

Write a simple code for binary search using function recursion in c programming language

```

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

    int a[10],i,n,m,c,l,u;

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

    printf("Enter the elements of the array: ");
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }

    printf("Enter the number to be search: ");
    scanf("%d",&m);

    l=0,u=n-1;
    c=binary(a,n,m,l,u);
    if(c==0)
        printf("Number is not found.");
    else
        printf("Number is found.");
}

```

```

        return 0;
    }

int binary(int a[],int n,int m,int l,int u){

    int mid,c=0;

    if(l<=u){
        mid=(l+u)/2;
        if(m==a[mid]){
            c=1;
        }
        else if(m<a[mid]){
            return binary(a,n,m,l,mid-1);
        }
        else
            return binary(a,n,m,mid+1,u);
    }
    else
        return c;
}

```

Sample output:

Enter the size of an array: 5

Enter the elements of the array: 8 9 10 11 12

Enter the number to be search: 8

Number is found.

CONCATENATION OF TWO STRINGS USING POINTER IN C PROGRAM

Concatenation of two strings using pointer in c programming language

```

#include<stdio.h>
int main(){
    int i=0,j=0;
    char *str1,*str2,*str3;
    puts("Enter first string");
    gets(str1);
    puts("Enter second string");
    gets(str2);
    printf("Before concatenation the strings are\n");
    puts(str1);
    puts(str2);
    while(*str1){
        str3[i++]=*str1++;
    }
}

```

```

    }
    while(*str2){
        str3[i++]=*str2++;
    }
    str3[i]='\0';
    printf("After concatenation the strings are\n");
    puts(str3);
    return 0;
}

```

COUNTING DIFFERENT CHARACTERS IN A STRING USING C PROGRAM

```

#include <stdio.h>
int isvowel(char chk);
int main(){
    char text[1000], chk;
    int count;
    count = 0;
    while((text[count] = getchar()) != '\n')
        count++;
    text[count] = '\0';
    count = 0;
    while ((chk = text[count]) != '\0'){
        if (isvowel(chk)){
            if((chk = text[++count]) && isvowel(chk)){
                putchar(text[count - 1]);
                putchar(text[count]);
                putchar('\n');
            }
        }
        else
            ++count;
    }
    return 0;
}
int isvowel(char chk){
    if(chk == 'a' || chk == 'e' || chk == 'i' || chk == 'o' || chk == 'u')
        return 1;
    return 0;
}

```

Alogrithm:

**

DISPLAY SOURCE CODE AS OUTPUT IN C PROGRAM

```
#include<stdio.h>
int main(){
    FILE *p;
    char ch;
    p=fopen("raja.c","r");
    while((ch=getc(p))!=-1)
        putchar(ch);
    fclose(p);
    return 0;
}
```

FIND POWER OF A NUMBER USING RECURSION USING C PROGRAM

Find power of a number using recursion using c program

```
#include<stdio.h>
int main(){
    int pow,num;
    long int res;
    long int power(int,int);
    printf("\nEnter a number: ");
    scanf("%d",&num);
    printf("\nEnter power: ");
    scanf("%d",&pow);
    res=power(num,pow);
    printf("\n%d to the power %d is: %ld",num,pow,res);
    return 0;
}
int i=1;
```

```
long int sum=1;
long int power(int num,int pow){
    if(i<=pow){
        sum=sum*num;
        power(num,pow-1);
    }
    else
        return sum;
}
```

FIND POWER OF A NUMBER USING C PROGRAM

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;
}
```

LINEAR SEARCH USING C PROGRAM

1. Write a simple code for linear search in c programming language
2. Wap a c program to search an element in an array using linear search

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

    int a[10],i,n,m,c=0;
```

```

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

printf("Enter the elements of the array: ");
for(i=0;i<=n-1;i++){
    scanf("%d",&a[i]);
}

printf("Enter the number to be search: ");
scanf("%d",&m);
for(i=0;i<=n-1;i++){
    if(a[i]==m){
        c=1;
        break;
    }
}
if(c==0)
    printf("The number is not in the list");
else
    printf("The number is found");

return 0;
}

```

Sample output:

```

Enter the size of an array: 5
Enter the elements of the array: 4 6 8 0 3
Enter the number to be search: 0
The number is found

```

C program for modular division of large number

1. C code for modular division of big numbers:
2. How to get modular division of two very large numbers larger or beyond than long int in c programming language

```

#include<stdio.h>
#include<string.h>
#define MAX 10000

int validate(char []);
int modulerDivision(char[],unsigned long);

int main(){

```

```

char dividend[MAX];
unsigned long int divisor,remainder;

printf("Enter dividend: ");
scanf("%s",dividend);
if(validate(dividend))
    return 0;

printf("Enter divisor: ");
scanf("%lu",&divisor);

remainder = modulerDivision(dividend,divisor);

printf("Modular division: %s %%
%lu = %lu",dividend,divisor,remainder);

return 0;
}

int validate(char num[]){
    int i=0;

    while(num[i]){
        if(num[i] < 48 || num[i]> 57){
            printf("Invalid positive integer: %s",num);
            return 1;
        }
        i++;
    }

    return 0;
}

int modulerDivision(char dividend[],unsigned long divisor){

    unsigned long temp=0;
    int i=0;

    while(dividend[i]){

        temp = temp*10 + (dividend[i] -48);
        if(temp>=divisor){
            temp = temp % divisor;
        }

        i++;
    }

    return temp;
}

```



```
}
```

Sample output:

Enter dividend: 123456789

Enter divisor: 56

Modular division: $123456789 \% 56 = 29$

Algorithm to find the modular division of large numbers:

Use array to store big numbers.

C program for multiplication of two binary numbers

C code for product of two binary numbers

C program for multiplication of two binary numbers

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

```
int binaryAddition(int,int);
```

```
int main(){
```

```
    long int binary1,binary2,multiply=0;
    int digit,factor=1;
```

```
    printf("Enter any first binary number: ");
    scanf("%ld",&binary1);
    printf("Enter any second binary number: ");
    scanf("%ld",&binary2);
```

```
    while(binary2!=0){
        digit =  binary2 %10;

        if(digit ==1){
            binary1=binary1*factor;
            multiply = binaryAddition(binary1,multiply);
        }
        else
            binary1=binary1*factor;
```

```
    binary2 = binary2/10;
    factor = 10;
```

```

    }

    printf("Product of two binary numbers: %ld",multiply);

    return 0;
}

int binaryAddition(int binary1,int binary2){

    int i=0,remainder = 0,sum[20];
    int binarySum=0;

    while(binary1!=0||binary2!=0){
        sum[i++] = (binary1 %10 + binary2 %10 + remainder ) % 2;
        remainder = (binary1 %10 + binary2 %10 + remainder ) / 2;
        binary1 = binary1/10;
        binary2 = binary2/10;
    }

    if(remainder!=0)
        sum[i++] = remainder;
    --i;
    while(i>=0)
        binarySum = binarySum*10 + sum[i--];

    return binarySum;
}

```

Sample output:

```

Enter any first binary number: 1101
Enter any second binary number: 11
Product of two binary numbers: 100111

```

Algorithm:

Rule of binary multiplication:

```

0 * 0 = 0
1 * 0 = 0
0 * 1 = 0
1 * 1 = 1

```

Q. what is the product of the binary numbers 0110 and 0011?

Answer: 0110 * 0011 = 10010

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.

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

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

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

Write a c program to check given string is palindrome number or not

```

#include<string.h>
#include<stdio.h>

```

```

int main(){
    char *str,*rev;
    int i,j;
    printf("\nEnter a string:");
    scanf("%s",str);
    for(i=strlen(str)-1,j=0;i>=0;i--,j++)
        rev[j]=str[i];
        rev[j]='\0';
    if(strcmp(rev,str))
        printf("\nThe string is not a palindrome");
    else
        printf("\nThe string is a palindrome");
    return 0;
}

```

Definition of Palindrome string:

A string is called palindrome if it symmetric. In other word a string is called palindrome if string remains same if its characters are reversed. For example: asdsa

If we will reverse it will remain same i.e. asdsa

Example of string palindrome: a,b, aa,aba,qwertrewq 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

$$F_n = F_{n-2} + F_{n-1}$$

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) = $1*2*3 \dots * n$
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

```

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
.....
.....
```

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

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;
}
```

Reverse any number using c program

Code 1:

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 2:

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 3:

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 4:

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

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

Code 1:

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;
        sum=sum*10+r;
        num=num/10;
    }
    printf("Sum of digits: %d",sum);
    return 0;
}

```

```

        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 2:

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 3:

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

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

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;
}

```

Sample output:
Enter any two integers: 5 10
Sum of two integers: 15

Algorithm:

In c ~ is 1's complement operator. This is equivalent to:
 $\sim a = -b + 1$
So, $a - \sim b - 1$
 $= a - (-b + 1) + 1$
 $= a + b - 1 + 1$
 $= a + b$

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

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;
}

```

Algorithm:

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

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;
}

```

FIND OUT GENERIC ROOT OF A NUMBER By C PROGRAM

C program for 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;
}
```

C code 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
```

Meaning of generic root:

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

So, generic root of 456 = 6

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

Write a c program to find out NCR factor of given number Or
C 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;
}
```

Algorithm:

In the mathematics nC_r has defined as
 ${}^nC_r = n! / ((n-r)!r!)$

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

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

```
#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

FIND PRIME FACTORS OF A NUMBER USING C PROGRAM

Prime factor of a number in c

```

#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)
            printf("\n%d is a prime factor",i);
    }
    i++;
}
return 0;
}

```

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

Alogrithm:

*

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

return 0;
}
```

C program for swapping of two numbers using pointers

```
#include<stdio.h>

int main(){

int a,b;
int *ptr a,*ptr b;
int *temp;

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

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

ptr a = &a;
ptr b = &b;

temp = ptr a;
```

```

        *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

```

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

```

Program to find largest of n numbers in c

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
```

Split number into digits in c programming

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

C program to count number of digits in a number

Code 1:
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

Code 2:
C code to count the total number of digit using for loop


```

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

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

    for(;num!=0;num=num/10)
        count++;

    printf("Total digits is:  %d",count);

    return 0;
}

```

Sample output:
Enter a number: 456
Total digits is: 3

Code 3:
Count the digits of a given number in c language using recursion

```

#include<stdio.h>

int countDigits(num);
int main(){
    int num,count;

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

    count = countDigits(num);

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

int countDigits(int num){
    static int count=0;

    if(num!=0){
        count++;
        countDigits(num/10);
    }

    return count;
}

```

Sample output:
Enter a number: 1234567

Total digits is: 7

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;
}

```

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.

SWAP TWO VARIABLES WITHOUT USING THIRD USING C PROGRAM VARIABLE

Swapping in c without temporary variable

Swap 2 numbers without using third variable in c

How to swap two numbers in c without using third variable

```
#include<stdio.h>  
  
int main(){  
    int a=5,b=10;  
  
    //process one  
    a=b+a;  
    b=a-b;  
    a=a-b;  
    printf("a= %d b= %d",a,b);
```

```

//process two
a=5;b=10;
a=a+b-(b=a);
printf("\na= %d  b=  %d",a,b);

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

//process four
a=5;b=10;
a=b~a-1;
b=a~b+1;
a=a~b+1;
printf("\na= %d  b=  %d",a,b);

//process five
a=5,b=10;
a=b+a,b=a-b,a=a-b;
printf("\na= %d  b=  %d",a,b);

return 0;
}

```

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:

Write a c program for finding gcd (greatest common divisor) of two given numbers

```
#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;
}
```

Other logic : **HCF (Highest common factor) 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;
}

```

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;
}

```

Write a c program for swapping of two arrays

Write a c program for swapping of two arrays

```
#include<stdio.h>
int main(){
    int a[10],b[10],c[10],i;
    printf("Enter First array->");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    printf("\nEnter Second array->");
    for(i=0;i<10;i++)
        scanf("%d",&b[i]);
    printf("Arrays before swapping");
    printf("\nFirst array->");
    for(i=0;i<10;i++){
        printf("%d",a[i]);
    }
    printf("\nSecond array->");
    for(i=0;i<10;i++){
        printf("%d",b[i]);
    }
    for(i=0;i<10;i++){
        //write any swapping technique
        c[i]=a[i];
        a[i]=b[i];
        b[i]=c[i];
    }
    printf("\nArrays after swapping");
    printf("\nFirst array->");
    for(i=0;i<10;i++){
        printf("%d",a[i]);
    }
    printf("\nSecond array->");
    for(i=0;i<10;i++){
        printf("%d",b[i]);
    }
    return 0;
}
```

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

Code 1:

1. Write a program 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 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.

Complex numbers definition

Complex numbers are two dimensional numbers i.e. it consists real part and imaginary part.

Complex numbers are written as $a + ib$

Where a and b are real numbers and $i^2 = -1$

Code 1:

1. How to use complex numbers in c
2. C language complex numbers

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

    printf("Enter any complex number in the format a+ib: ");
    scanf("%d+i%d",&a,&b);

    printf("Real part is: %d",a );
    printf("\nImaginary part is: %d",b);

    return 0;
}
```

Code 2:

1. C program to printf complex numbers

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

    printf("Enter the real part of complex number: ");
    scanf("%d",&a);

    printf("Enter the imaginary part of complex number: ");
    scanf("%d",&b);

    printf("\nComplex number is: %d%+di",a,b );

    return 0;
}
```

Operation on complex numbers:

Addition of complex number

$$(a + ib) + (c + id) = (a+c) + i(b+d)$$

Subtraction of complex numbers

$$(a + ib) - (c + id) = (a+c) + i(b-d)$$

Multiplication of complex numbers

$$(a + ib) * (c + id) = (ac-bd) + i(bc+ad)$$

Divison of complex numbers

$$(a + ib) / c + id = ((ac + bd) + i(bc-ad))/(c^2 + d^2)$$

Alogrithm:

WAPPING OF STRINGS USING C PROGRAM

Swapping of strings using c programming language

```
#include<stdio.h>
int main(){
    int i=0,j=0,k=0;
    char str1[20],str2[20],temp[20];
    puts("Enter first string");
    gets(str1);
    puts("Enter second string");
    gets(str2);
    printf("Before swaping the strings are\n");
    puts(str1);
    puts(str2);
    while(str1[i]!='\0'){
        temp[j++]=str1[i++];
    }
    temp[j]='\0';
    i=0,j=0;
    while(str2[i]!='\0'){
        str1[j++]=str2[i++];
    }
    str1[j]='\0';
```



```

i=0,j=0;
while(temp[i]!='\0'){
    str2[j++]=temp[i++];
}
str2[j]='\0';
printf("After swaping the strings are\n");
puts(str1);
puts(str2);
return 0;
}

```

C program for unit conversion

C code to convert one unit to other unit:

```

#include<stdio.h>
#include<math.h>

int main(){

    char fromUnit,toUnit;
    char *fUnit,*tUnit;
    long double fromValue,meterValue,toValue;
    int power =0;

    printf("To convert the 12 Inch to Foot\n");
    printf("enter the the unit in the format : dc12\n");

    printf("Ell: a\n");
    printf("Femi: b\n");
    printf("Foot: c\n");
    printf("Inch: d\n");
    printf("Light year: e\n");
    printf("Metre: f\n");
    printf("Mile: g\n");
    printf("Nano meter: h\n");
    printf("Pace: i\n");
    printf("Point: j\n");
    printf("Yard: k\n");
    printf("Mili meter: l\n");
    printf("Centi meter: m\n");
    printf("Deci meter: n\n");
    printf("Deca meter: o\n");
    printf("Hecto meter: p\n");
    printf("Kilo meter: q\n");

    scanf(" %c%c%Lf",&fromUnit,&toUnit,&fromValue);

```

```

switch(fromUnit){
    case 'a': meterValue = fromValue * 1.143; fUnit="ell";break;
    case 'b': meterValue = fromValue ; power = -15;
fUnit="fm"; break;
    case 'c': meterValue = fromValue * 0.3048; fUnit="ft";break;
    case 'd': meterValue = fromValue * 0.0254; fUnit="in";break;
    case 'e': meterValue = fromValue * 9.4607304725808; power =15;
fUnit="ly"; break;
    case 'f': meterValue = fromValue; fUnit="m"; break;
    case 'g': meterValue = fromValue * 1609.344; fUnit="mi"; break;
    case 'h': meterValue = fromValue; fUnit="nm"; power = -9; break;
    case 'i': meterValue = fromValue * 0.762 ; fUnit="pace"; break;
    case 'j': meterValue = fromValue * 0.000351450;
fUnit="pt"; break;
    case 'k': meterValue = fromValue * 0.9144; fUnit="yd";break;
    case 'l': meterValue = fromValue * 0.001; fUnit="mm";break;
    case 'm': meterValue = fromValue * 0.01; fUnit="cm";break;
    case 'n': meterValue = fromValue * 0.1; fUnit="deci
meter"; break;
    case 'o': meterValue = fromValue * 10; fUnit="deca meter"; break;
    case 'p': meterValue = fromValue * 100; fUnit="hm";break;
    case 'q': meterValue = fromValue * 1000; fUnit="km";break;
    default:
        printf("Invalid input"); exit(0);
}

switch(toUnit){
    case 'a': toValue = meterValue/1.143; tUnit="ell";break;
    case 'b': toValue = meterValue; tUnit="fm"; break;
    case 'c': toValue = meterValue/0.3048; tUnit="ft";break;
    case 'd': toValue = meterValue/0.0254; tUnit="in";break;
    case 'e': toValue = meterValue/9.4607304725808;
tUnit="ly"; break;
    case 'f': toValue = meterValue; tUnit="m";break;
    case 'g': toValue = meterValue/1609.344; tUnit="mi";break;
    case 'h': toValue = meterValue; tUnit="nm"; break;
    case 'i': toValue = meterValue/0.762; tUnit="pace";break;
    case 'j': toValue = meterValue/0.000351450; tUnit="pt";break;
    case 'k': toValue = meterValue/0.9144; tUnit="yd";break;
    case 'l': toValue = meterValue/0.001; tUnit="mm";break;
    case 'm': toValue = meterValue/0.01; tUnit="cm"; break;
    case 'n': toValue = meterValue/0.1; tUnit="deci meter";break;
    case 'o': toValue = meterValue/10; tUnit="deca meter";break;
    case 'p': toValue = meterValue/100; tUnit="hm"; break;
    case 'q': toValue = meterValue/1000; tUnit="km"; break;
    default:
        printf("Invalid input"); exit(0);
}

```

```

    if(power==0)
        printf("%.4Lf %s = %.4Lf %s",fromValue,fUnit,toValue,tUnit);
    else{
        while(tovalue > 10
            printf("%.4Lf %s = %.4Lf*10^%d
%s",fromValue,fUnit,toValue,power,tUnit);
        }

    return 0;
}

```

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

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;
}

```

Algorithm:

ASCII value of 'A' is 65 while 'a' is 97. Difference between them is 97 - 65 = 32

So if we will add 32 in the ASCII value of 'A' then it will be 'a' and if we will subtract 32 in ASCII value of 'a' it will be 'A'. It is true for all alphabets.

In general rule:

Upper case character = Lower case character - 32

Lower case character = Upper case character + 32

CONVERSION FROM UPPERCASE TO LOWER CASE USING C PROGRAM

Conversion from uppercase to lower case using c program

```
#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;
}
```

Algorithm:

ASCII value of 'A' is 65 while 'a' is 97. Difference between them is $97 - 65 = 32$

So if we will add 32 in the ASCII value of 'A' then it will be 'a' and if we subtract 32 in ASCII value of 'a' it will be 'A'. It is true for all alphabets.

In general rule:

Upper case character = Lower case character - 32

Lower case character = Upper case character + 32

SORTING OF STRING USING C PROGRAM

Program for sorting of string in c language

```
#include<stdio.h>
int main(){
    int i,j,n;
    char str[20][20],temp[20];
    puts("Enter the no. of string to be sorted");
    scanf("%d",&n);
    for(i=0;i<=n;i++){
        gets(str[i]);
    }
    for(i=0;i<=n;i++){
        for(j=i+1;j<=n;j++){
            if(strcmp(str[i],str[j])>0){
                strcpy(temp,str[i]);
                strcpy(str[i],str[j]);
                strcpy(str[j],temp);
            }
        }
    }
}
```

```
        for(j=i+1;j<=n;j++){
            if(strcmp(str[i],str[j])>0){
                strcpy(temp,str[i]);
                strcpy(str[i],str[j]);
                strcpy(str[j],temp);
            }
        }
        printf("The sorted string\n");
        for(i=0;i<=n;i++)
            puts(str[i]);
        return 0;
    }
}
```

Write a c program which prints initial of any name

C code which prints initial of any name

```
#include<stdio.h>
int main(){
    char str[20];
    int i=0;
    printf("Enter a string: ");
    gets(str);
    printf("%c",*str);
    while(str[i]!='\0'){
        if(str[i]==' '){
            i++;
            printf("%c",*(str+i));
        }
        i++;
    }
    return 0;
}
```

Sample output:

```
Enter a string: Robert De Niro
RDN
```

Write a c program to print the string from given character

```
#include<string.h>
#include<stdio.h>
int main(){
    char *p;
    char s[20],s1[1];
    printf("\nEnter a string: ");
    scanf("%[^\\n]",s);
    fflush(stdin);
    printf("\nEnter character: ");
    gets(s1);
    p=strpbrk(s,s1);
    printf("\nThe string from the given character is: %s",p);
    return 0;
}
```

Alogrithm:

**

Write a c program to reverse a string

Reverse a string in c without using temp
String reverse using strrev in c programming language

```
#include<stdio.h>
#include<string.h>
int main(){
    char str[50];
    char *rev;
    printf("Enter any string : ");
    scanf("%s",str);
    rev = strrev(str);

    printf("Reverse string is : %s",rev);

    return 0;
}
```

String reverse in c without using strrev
String reverse in c without using string function

How to reverse a string in c without using reverse function

```
#include<stdio.h>
int main(){
    char str[50];
    char rev[50];
    int i=-1,j=0;

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

    while(str[++i]!='\0');

    while(i>=0)
        rev[j++] = str[--i];

    rev[j]='\0';

    printf("Reverse of string is : %s",rev);

    return 0;
}
```

Sample output:

Enter any string : cquestionbank.blogspot.com

Reverse of string is : moc.topsgolb.knabnoitseuqc

Reverse a string in c using pointers

C program to reverse a string using pointers

```
#include<stdio.h>
int main(){
    char str[50];
    char rev[50];
    char *sptr = str;
    char *rptr = rev;
    int i=-1;

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

    while(*sptr){
        sptr++;
        i++;
    }

    while(i>=0){
        sptr--;
        *rptr = *sptr;
        rptr++;
    }
}
```

```

        --i;
    }

    *rptr='\0';

    printf("Reverse of string is : %s",rev);

    return 0;
}

```

Sample output:

Enter any string : Pointer

Reverse of string is : retnioP

Reverse a string using recursion in c

C code to reverse a string by recursion:

```

#include<stdio.h>
#define MAX 100
char* getReverse(char[]);

int main(){

    char str[MAX],*rev;

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

    rev = getReverse(str);

    printf("Reversed string is: %s",rev);
    return 0;
}

char* getReverse(char str[]){

    static int i=0;
    static char rev[MAX];

    if(*str){
        getReverse(str+1);
        rev[i++] = *str;
    }
}

```



```
    return rev;
}
```

Sample output:

Enter any string: mona
Reversed string is: anom

String concatenation in c without using strcat

String concatenation in c without using string functions

```
#include<stdio.h>

void stringConcat(char[],char[]);
int main(){

    char str1[100],str2[100];
    int compare;

    printf("Enter first string: ");
    scanf("%s",str1);

    printf("Enter second string: ");
    scanf("%s",str2);

    stringConcat(str1,str2);

    printf("String after concatenation: %s",str1);

    return 0;
}

void stringConcat(char str1[],char str2[]){
    int i=0,j=0;

    while(str1[i]!='\0'){
        i++;
    }

    while(str2[j]!='\0'){
        str1[i] = str2[j];
        i++;
        j++;
    }
}
```

```
    str1[i] = '\0';  
}
```

Sample output:

Enter first string: cquestionbank

Enter second string: @blogspot.com

String after concatenation: cquestionbank@blogspot.com

How to compare two strings in c without using strcmp

C program to compare two strings without using string functions

```
#include<stdio.h>  
  
int stringCompare(char[],char[];  
int main(){  
  
    char str1[100],str2[100];  
    int compare;  
  
    printf("Enter first string: ");  
    scanf("%s",str1);  
  
    printf("Enter second string: ");  
    scanf("%s",str2);  
  
    compare = stringCompare(str1,str2);  
  
    if(compare == 1)  
        printf("Both strings are equal.");  
    else  
        printf("Both strings are not equal");  
  
    return 0;  
}  
  
int stringCompare(char str1[],char str2[]){  
    int i=0,flag=0;
```

```

while(str1[i]!='\0' && str2[i]!='\0'){
    if(str1[i]!=str2[i]){
        flag=1;
        break;
    }
    i++;
}

if (flag==0 && str1[i]=='\0' && str2[i]=='\0')
    return 1;
else
    return 0;
}

```

Sample output:

Enter first string: cquestionbank.blogspot.com
Enter second string: cquestionbank.blogspot.com
Both strings are equal.

Alogrithm:

**

String copy without using strcpy in c

String copy without using strcpy in c programming language

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

```
void stringCopy(char[],char[]);
```

```
int main(){
```

```
    char str1[100],str2[100];
```

```
    printf("Enter any string: ");
```

```
    scanf("%s",str1);
```

```
    stringCopy(str1,str2);
```

```
    printf("After copying: %s",str2);
```

```

    return 0;
}

void stringCopy(char str1[],char str2[]){
    int i=0;

    while(str1[i]!='\0'){
        str2[i] = str1[i];
        i++;
    }

    str2[i]='\0';
}

```

Sample output:

Enter any string: cquestionbank.blogspot.com

After copying: cquestionbank.blogspot.com

ADDITION OF TWO MATRICES USING C PROGRAM

C program for addition of two matrices using arrays source code. Matrix addition in c language:

C code:

```

#include<stdio.h>
int main(){
    int a[3][3],b[3][3],c[3][3],i,j;
    printf("Enter the First matrix->");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);
    printf("\nEnter the Second matrix->");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%d",&b[i][j]);
    printf("\nThe First matrix is\n");
    for(i=0;i<3;i++){
        printf("\n");
        for(j=0;j<3;j++)
            printf("%d\t",a[i][j]);
    }
    printf("\nThe Second matrix is\n");
    for(i=0;i<3;i++){
        printf("\n");
    }
}

```

```

        for(j=0;j<3;j++)
            printf("%d\t",b[i][j]);
    }
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            c[i][j]=a[i][j]+b[i][j];
    printf("\nThe Addition of two matrix is\n");
    for(i=0;i<3;i++){
        printf("\n");
        for(j=0;j<3;j++)
            printf("%d\t",c[i][j]);
    }
    return 0;
}

```

Algorithm:

Addition of two matrices:

Rule: Addition of two matrices is only possible if both matrices are of same size.

Suppose two matrices A and B is of same size m X n

Sum of two matrices is defined as

$(A + B)_{ij} = A_{ij} + B_{ij}$
 Where $1 \leq i \leq m$ and $1 \leq j \leq n$

For example:

Suppose two matrices A and B of size of 2 X 3 is as follow:

SUBTRACTION OF TWO MATRICES USING C PROGRAM

```

#include<stdio.h>
int main(){
    int a[3][3],b[3][3],c[3][3],i,j;
    printf("Enter the First matrix->");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);
    printf("\nEnter the Second matrix->");
}

```

```

for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&b[i][j]);
printf("\nThe First matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
}
printf("\nThe Second matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",b[i][j]);
}
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        c[i][j]=a[i][j]-b[i][j];
printf("\nThe Subtraction of two matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",c[i][j]);
}
return 0;
}

```

Subtraction of two matrixes:

Rule: Subtraction of two matrixes is only possible if both matrixes are of same size.

Suppose two matrixes A and B is of same size m X n

Subtraction of two marixes is defined as

$$(A - B)_{ij} = A_{ij} - B_{ij}$$

Where $1 \leq i \leq m$ and $1 \leq j \leq n$

For example:

Suppose two matrixes A and B of size of 3 X 2 is as follow:

FIND OUT SUM OF DIAGONAL ELEMENTS OF A MATRIX USING

Sum of diagonal elements of a matrix in c

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

    int a[10][10],i,j,sum=0,m,n;

    printf("\nEnter the row and column of matrix: ");
    scanf("%d %d",&m,&n);

    printf("\nEnter the elements of matrix: ");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    printf("\nThe matrix is\n");

    for(i=0;i<m;i++){
        printf("\n");
        for(j=0;j<m;j++){
            printf("%d\t",a[i][j]);
        }
    }
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            if(i==j)
                sum=sum+a[i][j];
        }
    }
    printf("\n\nSum of the diagonal elements of a matrix is: %d",sum);

    return 0;
}
```

Sample output:

```
Enter the row and column of matrix: 3 3
Enter the elements of matrix: 2
3
5
6
```

```

7
9
2
6
7
The matrix is
2      3      5
6      7      9
2      6      7
Sum of the diagonal elements of a matrix is: 16

```

Alogrithm:

Sum of diagonal element of matrix:

$$A = \begin{pmatrix} \mathbf{A_{11}} & A_{12} & A_{13} \\ A_{21} & \mathbf{A_{22}} & A_{23} \\ A_{31} & A_{32} & \mathbf{A_{33}} \end{pmatrix}$$

Diagonal elements have been shown in the bold letter.

We can observer the properties any element A_{ij} will diagonal element if and only if $i = j$

MULTIPLICATION OF TWO MATRICES USING C PROGRAM

1. C code for matrix multiplication
2. C program for matrix multiplication
3. Write a program for matrix multiplication in c
4. How to multiply two matrixes in c
5. Matrix multiplication program in c language
6. Matrix multiplication in c using array

```
#include<stdio.h>
int main(){
    int a[5][5],b[5][5],c[5][5],i,j,k,sum=0,m,n,o,p;
    printf("\nEnter the row and column of first matrix");
    scanf("%d %d",&m,&n);
    printf("\nEnter the row and column of second matrix");
    scanf("%d %d",&o,&p);
    if(n!=o){
        printf("Matrix mutiplication is not possible");
        printf("\nColumn of first matrix must be same as row of second
matrix");
    }
    else{
        printf("\nEnter the First matrix->");
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                scanf("%d",&a[i][j]);
        printf("\nEnter the Second matrix->");
        for(i=0;i<o;i++)
            for(j=0;j<p;j++)
                scanf("%d",&b[i][j]);
        printf("\nThe First matrix is\n");
        for(i=0;i<m;i++){
            printf("\n");
            for(j=0;j<n;j++){
                printf("%d\t",a[i][j]);
            }
        }
        printf("\nThe Second matrix is\n");
        for(i=0;i<o;i++){
            printf("\n");
            for(j=0;j<p;j++){
                printf("%d\t",b[i][j]);
            }
        }
        for(i=0;i<m;i++)
            for(j=0;j<p;j++)
```

```

        c[i][j]=0;
    for(i=0;i<m;i++){ //row of first matrix
    for(j=0;j<p;j++){ //column of second matrix
        sum=0;
        for(k=0;k<n;k++)
            sum=sum+a[i][k]*b[k][j];
        c[i][j]=sum;
    }
}
printf("\nThe multiplication of two matrix is\n");
for(i=0;i<m;i++){
    printf("\n");
    for(j=0;j<p;j++){
        printf("%d\t",c[i][j]);
    }
}
return 0;
}

```

Alogrithm:

Multiplication of two matrixes:

Rule: Multiplication of two matrixes is only possible if first matrix has size $m \times n$ and other matrix has size $n \times r$. Where m , n and r are any positive integer.

Multiplication of two matrixes is defined as

$$[AB]_{i,j} = \sum_{s=1}^n A_{i,s} B_{s,j}$$

Where $1 \leq i \leq m$ and $1 \leq j \leq n$

For example:

Suppose two matrixes A and B of size of 2×2 and 2×3 respectively:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

Write a c program to find out transport of a matrix

C program for transpose of a matrix

C program to find transpose of given matrix

```
#include<stdio.h>
int main(){
    int a[10][10],b[10][10],i,j,k=0,m,n;
    printf("\nEnter the row and column of matrix");
    scanf("%d %d",&m,&n);
    printf("\nEnter the First matrix->");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
```

```

printf("\nThe matrix is\n");
for(i=0;i<m;i++){
    printf("\n");
    for(j=0;j<m;j++){
        printf("%d\t",a[i][j]);
    }
}
for(i=0;i<m;i++)
    for(j=0;j<n;j++)
        b[i][j]=0;
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        b[i][j]=a[j][i];
        printf("\n%d",b[i][j]);
    }
}
printf("\n\nTraspose of a matrix is -> ");
for(i=0;i<m;i++){
    printf("\n");
    for(j=0;j<m;j++){
        printf("%d\t",b[i][j]);
    }
}
return 0;
}

```

Alogrithm:

**

C program to find inverse of a matrix

How to find inverse of a matrix in c
 C code to find inverse of a matrix
 Inverse of a 3x3 matrix in c

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

```
int main(){
```

```

int a[3][3],i,j;
float determinant=0;

printf("Enter the 9 elements of matrix: ");
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&a[i][j]);

printf("\nThe matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
}

for(i=0;i<3;i++)
    determinant = determinant + (a[0][i]*(a[1][(i+1)%3]*a[2][(i+2)%3] -
a[1][(i+2)%3]*a[2][(i+1)%3]));

printf("\nInverse of matrix is: \n\n");
for(i=0;i<3;i++){
    for(j=0;j<3;j++)
        printf("%.2f\t",((a[(i+1)%3][(j+1)%3] * a[(i+2)%3][(j+2)%3]) -
(a[(i+1)%3][(j+2)%3]*a[(i+2)%3][(j+1)%3]))/ determinant);
    printf("\n");
}

return 0;
}

```

Enter the 9 elements of matrix: 3

5
2
1
5
8
3
9
2

The matrix is

| | | |
|---|---|---|
| 3 | 5 | 2 |
| 1 | 5 | 8 |
| 3 | 9 | 2 |

Inverse of matrix is:

| | | |
|-------|-------|------|
| 0.70 | -0.25 | 0.07 |
| -0.09 | -0.00 | 0.14 |

-0.34 0.25 -0.11

Alogrithm:

**

Lower triangular matrix in

C code to print or display lower triangular matrix

```
#include<stdio.h>
int main(){
    int a[3][3],i,j;
    float determinant=0;

    printf("Enter the 9 elements of matrix: ");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);

    printf("\nThe matrix is\n");
    for(i=0;i<3;i++){
        printf("\n");
        for(j=0;j<3;j++)
            printf("%d\t",a[i][j]);
    }

    printf("\nSetting zero in upper triangular matrix\n");
    for(i=0;i<3;i++){
        printf("\n");
```

```
        for(j=0;j<3;j++)
            if(i<=j)
                printf("%d\t",a[i][j]);
            else
                printf("%d\t",0);
    }

    return 0;
}
```

Enter the 9 elements of matrix: 1

2
3
4
5
6
7
8
9

The matrix is

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Setting zero in upper triangular matrix

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 0 | 5 | 6 |
| 0 | 0 | 9 |

Alogrithm:

**

Upper triangular matrix in c

C code to print or display upper triangular matrix

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

```

int a[3][3],i,j;
float determinant=0;

printf("Enter the 9 elements of matrix: ");
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&a[i][j]);

printf("\nThe matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
}

printf("\nSetting zero in upper triangular matrix\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        if(i>=j)
            printf("%d\t",a[i][j]);
        else
            printf("%d\t",0);
}

return 0;
}

```

Sample output:

Enter the 9 elements of matrix: 1

2
3
4
5
6
7
8
9

The matrix is

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Setting zero in upper triangular matrix

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 4 | 5 | 0 |
| 7 | 8 | 9 |

Alogrithm:

**

CREATE A FILE AND STORE DATA IN IT IN C PROGRAM

C program to create a file

C program to write to a file

C program to open a file

```
#include<stdio.h>
int main(){
    FILE *fp;
    char ch;
    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 find out the size and drive where file has stored of any given file?

```
#include "time.h"
#include "sys\stat.h"
#include "stdio.h"
int 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);
return 0;
}

```

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;
};

```

Here

(a) `st_dev`: It describe file has stored in which drive of your computer ,it returns a number.

(b) `st_mode`: It describes various modes of file like file is read only, write only, folder, character file etc.

(c) `st_size`: It tells the size of file in byte.

(d) `st_ctime`:It tells last data of modification of the file in date format.

Note: 65 is ASCII value of A .So after adding `status.st_dev` with 65 it will return appropriate drive name as in your computer.

Write a c program to find out the sum of series $1 + 2 + \dots + n$.

Sum of $1 + 2 + \dots + n$ series in c programming language

```

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

    int n,i;
    int sum=0;

```

```

printf("Enter the n i.e. max values of series: ");
scanf("%d",&n);

sum = (n * (n + 1)) / 2;

printf("Sum of the series: ");

for(i =1;i <= n;i++){
    if (i!=n)
        printf("%d + ",i);
    else
        printf("%d = %d ",i,sum);
}

return 0;
}

```

Sample output:

```

Enter the n i.e. max values of series: 5
Sum of the series: 1 + 2 + 3 + 4 + 5 = 15

```

Mathematical Formula:

Sum of the series $1 + 2 + 3 + \dots + n = n(n+1)/2$

Write a c program to find out the sum of series $1^2 + 2^2 + \dots + n^2$.

Sum of $1^2 + 2^2 + \dots + n^2$ series in c programming language

```

#include<stdio.h>

int main(){

    int n,i;
    int sum=0;

    printf("Enter the n i.e. max values of series: ");
    scanf("%d",&n);

    sum = (n * (n + 1) * (2 * n + 1 )) / 6;

    printf("Sum of the series : ");
}

```

```

for(i =1;i<=n;i++){
    if (i != n)
        printf("%d^2 + ",i);
    else
        printf("%d^2 = %d ",i,sum);
}

return 0;
}

```

Sample output:

Enter the n i.e. max values of series: 5
Sum of the series: $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$

Mathematical Formula:

Sum of the series $1^2 + 2^2 + 3^2 + \dots + n^2 =$
 $n(n+1)(2n+1)/6$

Write a c program to find out the sum of series $1^3 + 2^3 + \dots + n^3$

Write a c program or code to find out the sum of series $1^3 + 2^3 + \dots + n^3$ that is sum of cube of n natural numbers.

```

#include<stdio.h>
#include<math.h>

int main(){

    int n,i;
    int sum=0;

    printf("Enter the n i.e. max values of series: ");
    scanf("%d",&n);

    sum = pow(((n * (n + 1) ) / 2),2);

    printf("Sum of the series : ");

    for(i =1;i<=n;i++){
        if (i != n)
            printf("%d^3 + ",i);
        else
            printf("%d^3 = %d ",i,sum);
    }

    return 0;
}

```

```
}
```

Sample output:

Enter the n i.e. max values of series: 3
Sum of the series: $1^3 + 2^3 + 3^3 = 36$

Mathematical Formula:

Sum of the series $1^3 + 2^3 + 3^3 + \dots + n^3 = (n(n+1)/2)^2$

Write a c program to find out the sum of G.P series

Sum of GP series in c programming language

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

int main(){

    float a,r,i,tn;
    int n;
    float sum=0;

    printf("Enter the first number of the G.P. series: ");
    scanf("%f",&a);

    printf("Enter the total numbers in the G.P. series: ");
    scanf("%d",&n);

    printf("Enter the common ratio of G.P. series: ");
    scanf("%f",&r);

    sum = (a*(1 - pow(r,n+1)))/(1-r);
    tn = a * (1 -pow(r,n-1));

    printf("\n tn term of G.P.: %f",tn);
    printf("\n Sum of the G.P.: %f",sum);

    return 0;
}
```

Sample output:

Enter the first number of the G.P. series: 1
Enter the total numbers in the G.P. series: 5

Enter the common ratio of G.P. series: 2
tn term of G.P. : 16.000000
Sum of the G.P. : 63.000000

Definition of geometric progression (G.P.):

A series of numbers in which ratio of any two consecutive numbers is always a same number that is constant. This constant is called as common ratio.

Example of G.P. series:

2 4 8 16 32 64

Here common difference is 2 since ratio any two consecutive numbers for example 32 / 16 or 64/32 is 2.

Sum of G.P. series:

$$S_n = a(1-r^{n+1})/(1-r)$$

T_n term of G.P. series:

$$T_n = ar^{n-1}$$

Sum of infinite G.P. series:

$$S_n = \begin{matrix} a/(1-r) & \text{if } |r| < 1 \\ a/(r-1) & \text{if } r > 1 \end{matrix}$$

Write a c program to find the size of double without using sizeof operator

How to find size of double data type without using sizeof operator in c programming language

```
#include<stdio.h>

int main(){

    double *ptr = 0;

    ptr++;
    printf("Size of double data type:  %d",ptr);

    return 0;
}
```

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

How to find size of union data type without using sizeof operator in c programming language

```
#include<stdio.h>

union student{
    int roll;
    char name[100];
    float marks;
};

int main(){

    union student *ptr = 0;

    ptr++;
    printf("Size of the union student:  %d",ptr);

    return 0;
}
```

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

How to find size of structure data type without using sizeof operator in c programming language

```
#include<stdio.h>

struct student{
    int roll;
    char name[100];
    float marks;
};

int main(){

    struct student *ptr = 0;

    ptr++;
```

```
printf("Size of the structure student:  %d",ptr);  
  
return 0;  
}
```

C PROGRAM TO CALCULATE AREA OF A CIRCLE

C program for area of circle

```
#include <stdio.h>  
#define PI 3.141  
int main(){  
    float r, a;  
    printf("Radius: ");  
    scanf("%f", &r);  
    a = PI * r * r;  
    printf("%f\n", a);  
    return 0;  
}
```

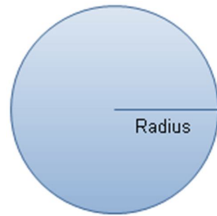
Mathematical formula for area of circle:

$$\text{Area} = \text{Pie} * \text{radius} * \text{radius}$$

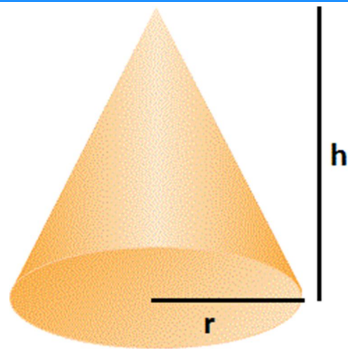
Here Pie is constant which is equal to

Pie = 22/7 or 3.14159265358979323846264338327950288419716939937510...

Radius is radius of the circle.



Write a c program to find the volume and surface area of cone



Formula of surface area of cone:

Surface_area = $\text{Pie} * r * (r + \sqrt{r^2 + h^2})$

Formula of volume of cone:

Volume = $\frac{1}{3} * \text{Pie} * r^2 * h$

Pie = 22/7 or 3.14159265358979323846264338327950288419716939937510...

Write a c program or code to find or calculate the volume and surface area of cone

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

int main(){

    float r,h;
    float surface_area,volume;

    printf("Enter size of radius and height of a cone : ");
    scanf("%f%f",&r,&h);
```

```

    surface_area = M_PI * r * (r + sqrt(r*r + h*h));
    volume = (1.0/3) * M_PI * r * r * h;

    printf("Surface area of cone is: %.3f",surface_area);
    printf("\nVolume of cone is : %.3f",volume);

    return 0;
}

```

Sample output:

```

Enter size of radius and height of a cone: 3 10
Surface area of cone is: 126.672
Volume of cone is: 94.248

```

Division of large numbers in c

1. C code for division of very big numbers.
2. How to get division of two very large numbers larger or beyond than long int in c programming language.

```

#include<stdio.h>
#include<string.h>
#define MAX 10000

int validate(char []);
char * division(char[],long);
long int remainder;

int main(){

    char dividend[MAX],*quotient;
    long int divisor;

    printf("Enter dividend: ");
    scanf("%s",dividend);
    if(validate(dividend))
        return 0;

    printf("Enter divisor: ");
    scanf("%ld",&divisor);

    quotient = division(dividend,divisor);

    while(*quotient)
        if(*quotient ==48)

```

```

        quotient++;
    else
        break;

    printf("Quotient: %s / %ld = %s",dividend,divisor,quotient);
    printf ("\nRemainder: %ld",remainder);
    return 0;
}

int validate(char num[]){
    int i=0;

    while(num[i]){
        if(num[i] < 48 || num[i]> 57){
            printf("Invalid positive integer: %s",num);
            return 1;
        }
        i++;
    }

    return 0;
}

char * division(char dividend[],long divisor){

    static char quotient[MAX];
    long temp=0;
    int i=0,j=0;

    while(dividend[i]){

        temp = temp*10 + (dividend[i] -48);
        if(temp<divisor){

            quotient[j++] = 48;

        }
        else{
            quotient[j++] = (temp / divisor) + 48;;
            temp = temp % divisor;
        }
        i++;
    }

    quotient[j] = '\0';
    remainder = temp;
    return quotient;
}

```

Sample output:

Enter dividend: 543109237823560187

Enter divisor: 456

Quotient: 543109237823560187 / 456 = 1191029030314824

Remainder: 443

Other program (source code):

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
#define MAX 10000

char * division(char [],unsigned long);
int main(){
    char a[MAX];
    unsigned long b;
    char *c;
    printf("Enter the dividend : ");
    scanf("%s",a);
    printf("Enter the divisor : ");
    scanf("%lu",&b);
    c = division(a,b);
    printf("\nQuotient of the division : ");
    printf("%s",c);
    return 0;
}

char * division(char a[],unsigned long b){
    static char c[MAX];
    int la;
    int i,k=0,flag=0;
    unsigned long temp=1,remainder;
    la=strlen(a);

    for(i=0;i<=la;i++){
        a[i] = a[i] - 48;
    }

    temp = a[0];
    remainder = a[0];
    for(i=1;i<=la;i++){
        if(b<=temp){
            c[k++] = temp/b;
            temp = temp % b;
            remainder = temp;
            temp =temp*10 + a[i];
            flag=1;
        }
    }
    if(flag==0)
        c[k++] = 0;
    return c;
}
```

```

    }
    else{
        reminder = temp;
        temp =temp*10 + a[i];
        if(flag==1)
            c[k++] = 0;
    }
}

for(i=0;i<k;i++){
    c[i]=c[i]+48;
}
c[i]= '\0';

printf("Reminder of division:  %lu  ",reminder);
return c;
}

```

Sample output:

Logic of the program Division of large numbers

As we know in c there are not any such data types which can store a very large numbers. For example we want to solve the expression:

Number 4928351200966342333331289107 is beyond the range of even long int or long double. Then question is how to store such a big numbers in c?

Multiplication of large numbers in c

1. Multiplication operation of very large numbers in c language

2. How to get multiplication of two very large numbers larger or beyond than long int in c programming language

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
#define MAX 10000

char * multiply(char [],char[]);
int main(){
    char a[MAX];
    char b[MAX];
    char *c;
    int la,lb;
    int i;
    printf("Enter the first number : ");
    scanf("%s",a);
    printf("Enter the second number : ");
    scanf("%s",b);
    printf("Multiplication of two numbers : ");
    c = multiply(a,b);
    printf("%s",c);
    return 0;
}

char * multiply(char a[],char b[]){
    static char mul[MAX];
    char c[MAX];
    char temp[MAX];
    int la,lb;
    int i,j,k=0,x=0,y;
    long int r=0;
    long sum = 0;
    la=strlen(a)-1;
    lb=strlen(b)-1;

    for(i=0;i<=la;i++){
        a[i] = a[i] - 48;
    }

    for(i=0;i<=lb;i++){
        b[i] = b[i] - 48;
    }
}
```

```

for(i=lb;i>=0;i--){
    r=0;
    for(j=la;j>=0;j--){
        temp[k++] = (b[i]*a[j] + r)%10;
        r = (b[i]*a[j]+r)/10;
    }
    temp[k++] = r;
    x++;
    for(y = 0;y<x;y++){
        temp[k++] = 0;
    }
}

k=0;
r=0;
for(i=0;i<la+lb+2;i++){
    sum =0;
    y=0;
    for(j=1;j<=lb+1;j++){
        if(i <= la+j){
            sum = sum + temp[y+i];
        }
        y += j + la + 1;
    }
    c[k++] = (sum+r) %10;
    r = (sum+r)/10;
}
c[k] = r;
j=0;
for(i=k-1;i>=0;i--){
    mul[j++]=c[i] + 48;
}
mul[j]='\0';
return mul;
}

```

Sample output of above code:

```

Enter the first number: 55555555
Enter the second number: 333333333
Multiplication of two numbers:
185185183314814815

```

Logic for multiplication of large numbers

As we know in c there are not any such data types which can store a very large numbers. For example we want to solve the expression:

```
55555555 * 333333333
```

Result of above expression is very big number which beyond the range of even long int or long double. Then question is how to store such a big numbers in c?

Solution is very simple i.e. using array. Above program has used same logic that is we are using as usual logic to multiply two numbers except instead of storing the data in the normal variables we are storing into the array.

C program to find power of a large number

C code for power of very big numbers:

How to get power of two very large numbers larger or beyond than long int in c programming language:

```
#include<stdio.h>
#include<string.h>
#define MAX 10000

char * multiply(char [],char[]);
int main(){
    char a[MAX];
    char *c;
    int i,n;
    printf("Enter the base number: ");
    scanf("%s",a);
    printf("Enter the power: ");
    scanf("%d",&n);

    printf("Power of the %s^%d: ",a,n);

    c = multiply(a,"1");
    for(i=0;i<n-1;i++)
        c = multiply(a,c);

    while(*c)
        if(*c == '0')
            c++;
        else
            break;

    printf("%s",c);
    return 0;
}

char * multiply(char num1[],char num2[]){
    static char mul[MAX];
```



```
char a[MAX];
char b[MAX];
char c[MAX];
char temp[MAX];
int la,lb;
int i=0,j,k=0,x=0,y;
long int r=0;
long sum = 0;

while(num1[i]){
    a[i] = num1[i];
    i++;
}
a[i]= '\0';

i=0;
while(num2[i]){
    b[i] = num2[i];
    i++;
}
b[i]= '\0';

la=strlen(a)-1;
lb=strlen(b)-1;

for(i=0;i<=la;i++){
    a[i] = a[i] - 48;
}

for(i=0;i<=lb;i++){
    b[i] = b[i] - 48;
}

for(i=lb;i>=0;i--){
    r=0;
    for(j=la;j>=0;j--){
        temp[k++] = (b[i]*a[j] + r)%10;
        r = (b[i]*a[j]+r)/10;
    }
    temp[k++] = r;
    x++;
    for(y = 0;y<x;y++){
        temp[k++] = 0;
    }
}

k=0;
r=0;
for(i=0;i<la+lb+2;i++){
```

```

    sum =0;
    y=0;
    for(j=1;j<=lb+1;j++){
        if(i <= la+j){
            sum = sum + temp[y+i];
        }
        y += j + la + 1;
    }
    c[k++] = (sum+r) %10;
    r = (sum+r)/10;
}
c[k] = r;
j=0;
for(i=k-1;i>=0;i--){
    mul[j++]=c[i] + 48;
}
mul[j]='\0';
return mul;
}

```

Sample output:

```

Enter the base number: 5
Enter the power: 100
Power of the 5^100: 78886090522101180541172856528278622
96732064351090230047702789306640625

```

Algorithm to find the power of large numbers:

Use array to store big numbers.

Write a c program which passes two dimension array to function

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");
    }
}

```

Algorithm:

**

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]);
            }
        }
    }
}

```

Write a c program which takes password from user

Write a c program which takes password from users

C source code for password:

```

#include<stdio.h>
#define MAX 500

int main(){

    char password[MAX];
    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: fgt67m,

Write a scanf function in c which accept sentence from user

Write a scanf function in c programming language which accept sentence from user

```

#include<stdio.h>
#define MAX 500

int main(){

    char arr[MAX];

    printf("Enter any sentence which can include spaces.\n");
    printf("To exit press enter key.\n");
    scanf("%[^\n]s",arr);

    printf("You had entered: \n");
    printf("%s",arr);

    return 0;
}

```

Sample output:

Enter any sentence which can include spaces.
To exit press enter key.
May I help you?
You had entered:
May I help you?

Write a scanf function in c which accept paragraph from user

Write a scanf function in c programming language which accept paragraph from users

```
#include<stdio.h>
#define MAX 500

int main(){

    char arr[MAX];

    printf("Enter any paragraph which can include spaces or new line.\n");
    printf("To exit press the tab key.\n");
    scanf("%[^\t]s",arr);

    printf("You had entered: \n");
    printf("%s",arr);

    return 0;
}
```

Sample output:

```
Enter any paragraph which can include spaces or new line.
To exit, press the tab key.
C is powerful language.
I am learning c from
cquestionbank.blogspot.com
```

```
You had entered:
C is powerful language.
I am learning c from
cquestionbank.blogspot.com
```

How to get the ASCII value of a character in c

Code to find the ASCII values of given character in c programming language:

```
#include<stdio.h>

int main(){
```

```

char c;

printf("Enter any character: ");
scanf("%c",&c);

printf("ASCII value of given character: %d",c);

return 0;
}

```

Sample output:

Enter any character: a

ASCII value of given character: 97

Print prime numbers between 1-300 using break and continue in c

Print prime numbers between 1-300 using break and continue in c

```

#include <math.h>
#include <stdio.h>
main(){
    int i, j;
    i = 2;
    while ( i < 300 ){
        j = 2;
        while ( j < sqrt(i) ){
            if ( i % j == 0 )
                break;
            else{
                ++j;
                continue;
            }
        }
        if ( j > sqrt(i) )
            printf("%d\t", i);
        ++i;
    }
    return 0;
}

```

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.

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.

Palindrome in c without using string function

```
#include<stdio.h>

int main(){
    char str[100];
    int i=0,j=-1,flag=0;

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

    while(str[++j]!='\0');
    j--;

    while(i<j)
        if(str[i++] != str[j--]){
            flag=1;
            break;
        }

    if(flag == 0)
        printf("The string is a palindrome");
    else
        printf("The string is not a palindrome");

    return 0;
}
```

Alogrithm:

**

C program to find out the sum of infinite G.P. series: geometric progression

Sum of infinite GP series in c programming language

```
#include<stdio.h>

int main(){

    float a,r;
    float sum=0;

    printf("Enter the first number of the G.P. series: ");
    scanf("%f",&a);

    printf("Enter the common ratio of G.P. series: ");
    scanf("%f",&r);

    if(1 > r)
        sum = a/(1-r);
    else
        sum = a/(r-1);

    printf("\nSum of the infinite G.P. series: %f",sum);

    return 0;
}
```

Sample output:

```
Enter the first number of the G.P. series: 1
Enter the common ratio of G.P. series: .5
```

```
Sum of the infinite G.P. series: 2.000000
```

```
Enter the first number of the G.P. series: 5
Enter the common ratio of G.P. series: 2
```

```
Sum of the infinite G.P. series: 5.000000
```

Definition of geometric progression (G.P.):

A series of numbers in which ratio of any two consecutive numbers is always a same number that is constant. This constant is called as common ratio.

Example of G.P. series:

2 4 8 16 32 64

Here common difference is 2 since ratio any two consecutive numbers for example $32 / 16$ or $64/32$ is 2.

Sum of G.P. series:

$$S_n = a(1-r^{n+1})/(1-r)$$

T_n term of G.P. series:

$$T_n = ar^{n-1}$$

Sum of infinite G.P. series:

$$S_n = \begin{matrix} a/(1-r) & \text{if } 1 > r \\ a/(r-1) & \text{if } r > 1 \end{matrix}$$

INSERT AN ELEMENT IN AN ARRAY AT DESIRED POSITION USING C PROGRAM

How to insert or add an element in the array at specific or desired position by using c programming language?
Source code is as follow:

```
#include<stdio.h>
int main(){
    int a[50],size,num,i,pos,temp;
    printf("\nEnter size of the array: ");
    scanf("%d",&size);
    printf("\nEnter %d elements in to the array: ",size);
    for(i=0;iscanf("%d",&a[i]);
printf("\nEnter position and number to insert: ");
    scanf("%d %d",&pos,&num);
    i=0;
    while(i!=pos-1)
        i++;
    temp=size++;
    while(i{
        a[temp]=a[temp-1];
        temp--;
    }
```

```
        a[i]=num;
        for(i=0;i<size;i++)printf(" %d",a[i]);
return 0;
}
```

FIND OUT LARGEST NUMBER IN AN ARRAY USING C PROGRAM

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);
    for(i=0;i<size;i++)
        scanf("%d",&a[i]);
    big=a[0];
    for(i=1;i<size;i++){
        if(big<a[i])
            big=a[i];
    }
    printf("\nBiggest: %d",big);
    return 0;
}
```

Alogrithm:

**

Write a c program to reverse a string

Reverse a string in c without using temp
String reverse using strrev in c programming language

```
#include<stdio.h>
#include<string.h>
int main(){
    char str[50];
    char *rev;
    printf("Enter any string : ");
    scanf("%s",str);
```

```

    rev = strrev(str);

    printf("Reverse string is : %s",rev);

    return 0;
}

```

String reverse in c without using strrev
String reverse in c without using string function
How to reverse a string in c without using reverse function

```

#include<stdio.h>
int main(){
    char str[50];
    char rev[50];
    int i=-1,j=0;

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

    while(str[++i]!='\0');

    while(i>=0)
        rev[j++] = str[--i];

    rev[j]='\0';

    printf("Reverse of string is : %s",rev);

    return 0;
}

```

Sample output:
Enter any string : cquestionbank.blogspot.com
Reverse of string is : moc.topsgolb.knabnoitseugc

Reverse a string in c using pointers
C program to reverse a string using pointers

```

#include<stdio.h>
int main(){
    char str[50];
    char rev[50];
    char *sptr = str;
    char *rprr = rev;
    int i=-1;

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

```

```

while(*sptr){
    sptr++;
    i++;
}

while(i>=0){
    sptr--;
    *rptr = *sptr;
    rptr++;
    --i;
}

*rptr='\0';

printf("Reverse of string is : %s",rev);

return 0;
}

```

Sample output:
Enter any string : Pointer
Reverse of string is : retnioP

C program to find out the sum of infinite G.P. series: geometric progression

Sum of infinite GP series in c programming language

```

#include<stdio.h>

int main(){

    float a,r;
    float sum=0;

    printf("Enter the first number of the G.P. series: ");
    scanf("%f",&a);

    printf("Enter the common ratio of G.P. series: ");
    scanf("%f",&r);

    if(1 > r)
        sum = a/(1-r);
    else
        sum = a/(r-1);
}

```

```
printf("\nSum of the infinite G.P. series: %f",sum);  
  
return 0;  
}
```

Sample output:

Enter the first number of the G.P. series: 1
Enter the common ratio of G.P. series: .5

Sum of the infinite G.P. series: 2.000000

Enter the first number of the G.P. series: 5
Enter the common ratio of G.P. series: 2

Sum of the infinite G.P. series: 5.000000

Definition of geometric progression (G.P.):

A series of numbers in which ratio of any two consecutive numbers is always a same number that is constant. This constant is called as common ratio.

Example of G.P. series:

2 4 8 16 32 64

Here common difference is 2 since ratio any two consecutive numbers for example 32 / 16 or 64/32 is 2.

Sum of G.P. series:

$S_n = a(1-r^{n+1})/(1-r)$

T_n term of G.P. series:

$T_n = ar^{n-1}$

Sum of infinite G.P. series:

$S_n = a/(1-r) \quad \text{if } |r| < 1$
 $= a/(r-1) \quad \text{if } r > 1$

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

$$F_n = F_{n-2} + F_{n-1}$$

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.

QUICK SORT USING C PROGRAM

Source code of simple quick sort implementation using array ascending order in c programming language

```
#include<stdio.h>

void quicksort(int [10],int,int);

int main(){
    int x[20],size,i;

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

    printf("Enter %d elements: ",size);
    for(i=0;i<size;i++)
        scanf("%d",&x[i]);

    quicksort(x,0,size-1);

    printf("Sorted elements: ");
    for(i=0;i<size;i++)
        printf(" %d",x[i]);

    return 0;
}

void quicksort(int x[10],int first,int last){
    int pivot,j,temp,i;

    if(first<last){
        pivot=first;
        i=first;
        j=last;

        while(i<j){
            while(x[i]<=x[pivot]&& i<last)
                i++;
            while(x[j]>x[pivot])
                j--;
            if(i<j){
                temp=x[i];
```

```

        x[i]=x[j];
        x[j]=temp;
    }

    temp=x[pivot];
    x[pivot]=x[j];
    x[j]=temp;
    quicksort(x,first,j-1);
    quicksort(x,j+1,last);
}
}

```

Output:

```

Enter size of the array: 5
Enter 5 elements: 3 8 0 1 2
Sorted elements: 0 1 2 3 8

```

Merge sort program in c

Source code of simple merge sort implementation using array in ascending order in c programming language

```

#include<stdio.h>
#define MAX 50

void mergeSort(int arr[],int low,int mid,int high);
void partition(int arr[],int low,int high);

int main(){
    int merge[MAX],i,n;

    printf("Enter the total number of elements: ");
    scanf("%d",&n);

    printf("Enter the elements which to be sort: ");
    for(i=0;i<n;i++){
        scanf("%d",&merge[i]);
    }

    partition(merge,0,n-1);

    printf("After merge sorting elements are: ");
    for(i=0;i<n;i++){

```

```

        printf("%d ",merge[i]);
    }
    return 0;
}

void partition(int arr[],int low,int high){

    int mid;

    if(low<high){
        mid=(low+high)/2;
        partition(arr,low,mid);
        partition(arr,mid+1,high);
        mergeSort(arr,low,mid,high);
    }
}

void mergeSort(int arr[],int low,int mid,int high){

    int i,m,k,l,temp[MAX];

    l=low;
    i=low;
    m=mid+1;

    while((l<=mid)&&(m<=high)){

        if(arr[l]<=arr[m]){
            temp[i]=arr[l];
            l++;
        }
        else{
            temp[i]=arr[m];
            m++;
        }
        i++;
    }

    if(l>mid){
        for(k=m;k<=high;k++){
            temp[i]=arr[k];
            i++;
        }
    }
    else{
        for(k=l;k<=mid;k++){
            temp[i]=arr[k];
            i++;
        }
    }
}

```

```

    }

    for(k=low;k<=high;k++){
        arr[k]=temp[k];
    }
}

```

Sample output:

```

Enter the total number of elements: 5
Enter the elements which to be sort: 2 5 0 9 1
After merge sorting elements are: 0 1 2 5 9

```

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) = $1*2*3 \dots * n$

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

Note: Factorial of zero = 1

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 divided 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:

Write a c program for finding gcd (greatest common divisor) of two given numbers

```
#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;
}
```

Other logic : HCF (Highest common factor) 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;
}
```

```
}
```

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;
}
```

SELECTION SORT USING C PROGRAM

Source code of simple Selection sort implementation using array ascending order in c programming language

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

```

int main(){

    int s,i,j,temp,a[20];

    printf("Enter total elements: ");
    scanf("%d",&s);

    printf("Enter %d elements: ",s);
    for(i=0;i<s;i++)
        scanf("%d",&a[i]);

    for(i=0;i<s;i++){
        for(j=i+1;j<s;j++){
            if(a[i]>a[j]){
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }

    printf("After sorting is: ");
    for(i=0;i<s;i++)
        printf(" %d",a[i]);

    return 0;
}

```

Output:

Enter total elements: 5

Enter 5 elements: 4 5 0 21 7

The array after sorting is: 0 4 5 7 21

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

How to test palindrome in c++

How to test a number is palindrome or not in c++

```
#include<iostream>

int checkPalindrome(int);
int main(){
    int num,sum;

    cout << "Enter a number: ";
    cin >> num;

    sum = checkPalindrome(num);

    if(num==sum)
        cout << num << " is a palindrome";
    else
        cout << num << " is not a palindrome";

    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;
}
```

Algorithm:

**

Palindrome number in c++

Palindrome number in c++

```
#include<iostream>
int main(){
    int num,r,sum=0,temp;

    cout << "Enter a number: ";
    cin >> num;

    for(temp=num;num!=0;num=num/10){
        r=num%10;
        sum=sum*10+r;
    }
    if(temp==sum)
        cout << temp << " is a palindrome";
    else
        cout << temp << " is not a palindrome";

    return 0;
}
```

Algorithm:

**

Program of palindrome number in c++

Program of palindrome number in c++

```
#include<iostream>
int main(){
    int num,r,sum,temp;
    int min,max;

    cout << "Enter the minimum range: ";
    cin >> min;

    cout << "Enter the maximum range: ";
    cin >> max;

    cout << "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)
            cout << num << " ";
    }
    return 0;
}

```

Check the given number is palindrome number or not by c++ program

```

#include<iostream>
int main(){
    int num,r,sum=0,temp;

    cout << "Enter a number: ";
    cin >> num;

    for(temp=num;num!=0;num=num/10){
        r=num%10;
        sum=sum*10+r;
    }
    if(temp==sum)
        cout << temp << " is a palindrome";
    else
        cout << " is not a palindrome";

    return 0;
}

```

C PROGRAM FOR INSERTION SORT

Source code of simple insertion sort implementation using array in ascending order in c programming language

```

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

```



```

int i,j,s,temp,a[20];

printf("Enter total elements: ");
scanf("%d",&s);

printf("Enter %d elements: ",s);
for(i=0;i<s;i++)
    scanf("%d",&a[i]);

for(i=1;i<s;i++){
    temp=a[i];
    j=i-1;
    while((temp<a[j])&&(j>=0)){
        a[j+1]=a[j];
        j=j-1;
    }
    a[j+1]=temp;
}

printf("After sorting: ");
for(i=0;i<s;i++)
    printf(" %d",a[i]);

return 0;
}

```

Output:

```

Enter total elements: 5
Enter 5 elements: 3 7 9 0 2
After sorting: 0 2 3 7 9

```

FIND OUT SECOND LARGEST NUMBER IN AN UNSORTED ARRAY USING C PROGRAM

C program to find the second largest element in an array

```

#include<stdio.h>
int main(){
    int a[50],size,i,j=0,big,secondbig;
    printf("Enter the size of the array: ");
    scanf("%d",&size);
    printf("Enter %d elements in to the array: ", size);
    for(i=0;i<size;i++)
        scanf("%d",&a[i]);

    big=a[0];

```

```

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

secondbig=a[size-j-1];
for(i=1;i<size;i++){
    if(secondbig <a[i] && j != i)
        secondbig =a[i];
}

printf("Second biggest: %d", secondbig);
return 0;
}

```

Sample output:

Enter the size of the array: 5
Enter 5 elements in to the array: 5 3 2 1 0
Second biggest: 3

C program to find largest and smallest number in an array

C code to find largest and smallest number in an array

```

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

    printf("\nEnter the size of the array: ");
    scanf("%d",&size);
    printf("\nEnter %d elements in to the array: ", size);
    for(i=0;i<size;i++)
        scanf("%d",&a[i]);

    big=a[0];

```

```

    for(i=1;i<size;i++){
        if(big<a[i])
            big=a[i];
    }
    printf("Largest element: %d",big);

    small=a[0];
    for(i=1;i<size;i++){
        if(small>a[i])
            small=a[i];
    }
    printf("Smallest element: %d",small);

    return 0;
}

```

Sample Output:

```

Enter the size of the array: 4
Enter 4 elements in to the array: 2 7 8 1
Largest element: 8
Smallest element: 1

```

Algorithm:

**

DELETE ELEMENT FROM AN ARRAY AT DESIRED POSITION USING C

Write a program (wap) to delete an element at desired position from an array in c language

```

#include<stdio.h>
int main(){
    int a[50],i,pos,size;
    printf("\nEnter size of the array: ");
    scanf("%d",&size);

    printf("\nEnter %d elements in to the array: ",size);
    for(i=0;i<size;i++)
        scanf("%d",&a[i]);

    printf("\nEnter position where to delete: ");
    scanf("%d",&pos);

```

```

i=0;
while(i!=pos-1)
    i++;
while(i<10){
    a[i]=a[i+1];
    i++;
}

size--;
for(i=0;i<size;i++)
    printf(" %d",a[i]);

return 0;
}

```

INSERT AN ELEMENT IN AN ARRAY AT DESIRED POSITION USING C PROGRAM

How to insert or add an element in the array at specific or desired posting by using c programming language? Source code is as follow:

```

#include<stdio.h>
int main(){
    int a[50],size,num,i,pos,temp;
    printf("\nEnter size of the array: ");
    scanf("%d",&size);
    printf("\nEnter %d elements in to the array: ",size);
    for(i=0;iscanf("%d",&a[i]);
printf("\nEnter position and number to insert: ");
    scanf("%d %d",&pos,&num);
    i=0;
    while(i!=pos-1)
        i++;
    temp=size++;
    while(i{
        a[temp]=a[temp-1];
        temp--;
    }
    a[i]=num;
    for(i=0;iprintf(" %d",a[i]);
return 0;
}

```

Write a c program to find out second smallest element of an unsorted array

C program to find the second smallest element in an array

```
#include<stdio.h>
int main(){
    int a[50],size,i,j=0,small,secondsmall;
    printf("Enter the size of the array: ");
    scanf("%d",&size);
    printf("Enter %d elements in to the array: ", size);
    for(i=0;i<size;i++){
        scanf("%d",&a[i]);

        small=a[0];
        for(i=1;i<size;i++){
            if(small>a[i]){
                small=a[i];
                j = i;
            }
        }

        secondsmall=a[size-j-1];
        for(i=1;i<size;i++){
            if(secondsmall > a[i] && j != i)
                secondsmall =a[i];
        }

        printf("Second smallest: %d", secondsmall);
        return 0;
    }
}
```

Enter the size of the array: 5
Enter 5 elements in to the array: 5 7 3 2 6
Second smallest: 3

REMOVE DUPLICATE ELEMENTS IN AN ARRAY USING C PROGRAM

```
#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);
for(i=0;i<n;i++)
    scanf("%d",&arr[i]);
size=n;
p=arr;
for(i=0;i<size;i++){
    for(j=0;j<size;j++){
        if(i==j){
            continue;
        }
        else if(*(p+i)==*(p+j)){
            k=j;
            size--;
            while(k < size){
                *(p+k)=*(p+k+1);
                k++;
            }
            j=0;
        }
    }
}
printf("\nThe array after removing duplicates is: ");
for(i=0;i < size;i++){
    printf(" %d",arr[i]);
}
return 0;
}

```

SELECTION SORT USING C PROGRAM

Source code of simple Selection sort implementation using array ascending order in c programming language

```

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

    int s,i,j,temp,a[20];

    printf("Enter total elements: ");
    scanf("%d",&s);

    printf("Enter %d elements: ",s);
    for(i=0;i<s;i++)

```

```

scanf("%d",&a[i]);

for(i=0;i<s;i++){
    for(j=i+1;j<s;j++){
        if(a[i]>a[j]){
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
}

printf("After sorting is: ");
for(i=0;i<s;i++)
    printf(" %d",a[i]);

return 0;
}

```

Output:

Enter total elements: 5

Enter 5 elements: 4 5 0 21 7

The array after sorting is: 0 4 5 7 21

FIND SUM OF DIGITS OF A NUMBER USING RECURSION USING C PROGRAM

Sum of digits in c using recursion

```

#include<stdio.h>
int main(){
    int num,x;
    clrscr();
    printf("\nEnter a number: ");
    scanf("%d",&num);
    x=findsum(num);
    printf("Sum of the digits of %d is: %d",num,x);
    return 0;
}

int r,s;
int findsum(int n){
    if(n){
        r=n%10;
        s=s+r;
        findsum(n/10);
    }
    else

```

```
        return s;
    }
}
```

FIND GCD OF A NUMBER USING RECURSION IN C PROGRAM

Find gcd of a number using recursion in c program

```
#include<stdio.h>
int main(){
    int n1,n2,gcd;
    printf("\nEnter two numbers: ");
    scanf("%d %d",&n1,&n2);
    gcd=findgcd(n1,n2);
    printf("\nGCD of %d and %d is: %d",n1,n2,gcd);
    return 0;
}

int findgcd(int x,int y){
    while(x!=y){
        if(x>y)
            return findgcd(x-y,y);
        else
            return findgcd(x,y-x);
    }
    return x;
}
```

Merge sort program in c

Source code of simple merge sort implementation using array in ascending order in c programming language

```
#include<stdio.h>
#define MAX 50

void mergeSort(int arr[],int low,int mid,int high);
void partition(int arr[],int low,int high);

int main(){
```



```

    int merge[MAX],i,n;

    printf("Enter the total number of elements: ");
    scanf("%d",&n);

    printf("Enter the elements which to be sort: ");
    for(i=0;i<n;i++){
        scanf("%d",&merge[i]);
    }

    partition(merge,0,n-1);

    printf("After merge sorting elements are: ");
    for(i=0;i<n;i++){
        printf("%d ",merge[i]);
    }

    return 0;
}

void partition(int arr[],int low,int high){

    int mid;

    if(low<high){
        mid=(low+high)/2;
        partition(arr,low,mid);
        partition(arr,mid+1,high);
        mergeSort(arr,low,mid,high);
    }
}

void mergeSort(int arr[],int low,int mid,int high){

    int i,m,k,l,temp[MAX];

    l=low;
    i=low;
    m=mid+1;

    while((l<=mid)&&(m<=high)){

        if(arr[l]<=arr[m]){
            temp[i]=arr[l];
            l++;
        }
        else{
            temp[i]=arr[m];
            m++;
        }
    }
}

```

```

    }
    i++;
}

if(l>mid){
    for(k=m;k<=high;k++){
        temp[i]=arr[k];
        i++;
    }
}
else{
    for(k=l;k<=mid;k++){
        temp[i]=arr[k];
        i++;
    }
}

for(k=low;k<=high;k++){
    arr[k]=temp[k];
}
}
}

```

Sample output:

```

Enter the total number of elements: 5
Enter the elements which to be sort: 2 5 0 9 1
After merge sorting elements are: 0 1 2 5 9

```

FIND FACTORIAL OF A NUMBER USING RECURSION IN C PROGRAM

1. Factorial program by recursion in c
2. Factorial program in c using recursion
3. C program to calculate factorial using recursion
4. Recursive function for factorial in c

```

#include<stdio.h>
int fact(int);
int main(){
    int num,f;
    printf("\nEnter a number: ");
    scanf("%d",&num);
    f=fact(num);

```

```
printf("\nFactorial of %d is: %d",num,f);
return 0;
}

int fact(int n){
    if(n==1)
        return 1;
    else
        return(n*fact(n-1));
}
```

QUICK SORT USING C PROGRAM

Source code of simple quick sort implementation using array ascending order in c programming language

```
#include<stdio.h>

void quicksort(int [10],int,int);

int main(){
    int x[20],size,i;

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

    printf("Enter %d elements: ",size);
    for(i=0;i<size;i++)
        scanf("%d",&x[i]);

    quicksort(x,0,size-1);

    printf("Sorted elements: ");
    for(i=0;i<size;i++)
        printf(" %d",x[i]);

    return 0;
}

void quicksort(int x[10],int first,int last){
    int pivot,j,temp,i;

    if(first<last){
        pivot=first;
        i=first;
```

```

j=last;

while(i<j){
    while(x[i]<=x[pivot]&& i<last)
        i++;
    while(x[j]>x[pivot])
        j--;
    if(i<j){
        temp=x[i];
        x[i]=x[j];
        x[j]=temp;
    }

    temp=x[pivot];
    x[pivot]=x[j];
    x[j]=temp;
    quicksort(x,first,j-1);
    quicksort(x,j+1,last);
}
}

```

Output:
Enter size of the array: 5
Enter 5 elements: 3 8 0 1 2
Sorted elements: 0 1 2 3 8

Write a c program to find the area of an equilateral triangle

C program for area of equilateral triangle

Formula of area of equilateral triangle:

Area = $(\sqrt{3})/4 * a^2$

C code:

```

#include<stdio.h>
#include<math.h>

int main(){

    float a;
    float area;

```

```
printf("Enter size of side of the equilateral triangle : ");
scanf("%f",&a);

area = sqrt(3)/4*(a*a);

printf("Area of equilateral triangle is: %.3f",area);

return 0;
}
```

Sample output:

Enter size of side of the equilateral triangle: 5
Area of equilateral triangle is: 10.825

Write a c program to find the size of int without using sizeof operator

How to find size of integer data type without using sizeof operator in c programming language

```
#include<stdio.h>

int main(){

    int *ptr = 0;

    ptr++;
    printf("Size of int data type: %d",ptr);

    return 0;
}
```

Write a c program to find the area of a right angled triangle
C program for area of right angled triangle

Formula of area of right angled triangle:

Area = (1/2) * height * width

C code:

```
#include<stdio.h>

int main(){

    float h,w;
    float area;

    printf("Enter height and width of the right angled triangle : ");
    scanf("%f%f",&h,&w);

    area = 0.5 * h * w;

    printf("Area of right angled triangle is: %.3f",area);

    return 0;
}
```

Sample output:

```
Enter height and width of the right angled triangle: 10 5
Area of right angled triangle is: 25.000
```

Write a c program to find the area of a trapezium
C program for area of a trapezium

Formula of area of trapezium:

$$\text{Area} = (1/2) * (a + b) * h$$

C code:

```
#include<stdio.h>

int main(){

    float b1,b2,h;
    float area;

    printf("Enter the size of two bases and height of the trapezium : ");
    scanf("%f%f%f",&b1,&b2,&h);
```

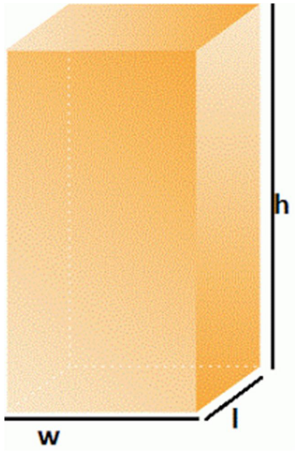
```
area = 0.5 * ( b1 + b2 ) * h ;  
  
printf("Area of trapezium is: %.3f",area);  
  
return 0;  
}
```

Sample output:

Enter the size of two bases and height of the trapezium: 5 8 3
Area of trapezium is: 19.500

Write a c program to find the volume and surface area of cuboids

C program for area of a cuboids



Formula of surface area of cuboids:
 $\text{Surface_area} = 2 * (w * l + l * h + h * w)$

Formula of volume of cuboids:
 $\text{Volume} = w * l * h$

Space diagonal of cuboids:

$\text{Space_diagonal} = \sqrt{w^2 + l^2 + h^2}$

C code:

```
#include<stdio.h>

int main(){

    float w,l,h;
    float surface_area,volume,space_diagonal;

    printf("Enter size of width, length and height of a cuboids : ");
    scanf("%f%f%f",&w,&l,&h);

    surface_area = 2*(w*l + l*h + h*w);
    volume = w * l * h;
    space_diagonal = sqrt(w*w + l*l + h*h);

    printf("Surface area of cuboids is: %.3f",surface_area);
    printf("\nVolume of cuboids is : %.3f",volume);
    printf("\nSpace diagonal of cuboids is : %.3f",space_diagonal);

    return 0;
}
```

Sample output:

```
Enter size of width, length and height of cuboids: 5 10 4
Surface area of cuboids is: 220.000
Volume of cuboids is: 200.000
Space diagonal of cuboids is: 11.874
```

Write a c program to find the area of a rectangle

C program for area of a rectangle

Formula of area of rectangle:

Area = length * width

C code:

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



```

int main(){

    float l,w;
    float area;

    printf("Enter size of each sides of the rectangle : ");
    scanf("%f%f",&l,&w);

    area = l * w;
    printf("Area of rectangle is: %.3f",area);

    return 0;
}

```

Sample output:

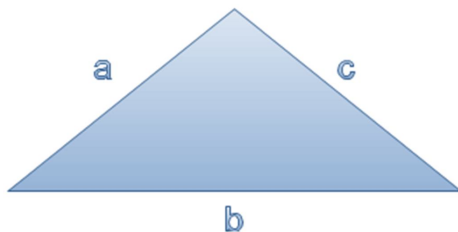
```

Enter size of each sides of the rectangle: 5.2 20
Area of rectangle is: 104.000

```

Write a c program to find the area of a triangle

C program for area of a triangle



Formula of area of any triangle:

Area = $\sqrt{s(s-a)(s-b)(s-c)}$

Where $s = (a + b + c) / 2$

C code:

```

#include<stdio.h>

```

```
#include<math.h>

int main(){

    float a,b,c;
    float s,area;

    printf("Enter size of each sides of triangle");
    scanf("%f%f%f",&a,&b,&c);

    s = (a+b+c)/2;
    area = sqrt(s*(s-a)*(s-b)*(s-c));

    printf("Area of triangle is: %.3f",area);

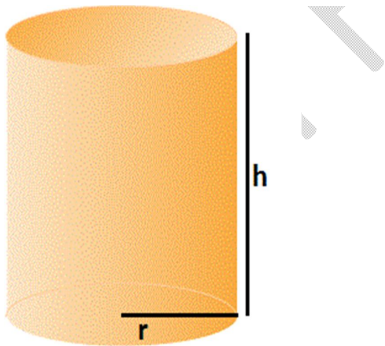
    return 0;
}
```

Sample output:

Enter size of each sides of the triangle: 2 4 5
Area of triangle is: 3.800

Write a c program to find the volume and surface area of cylinder

C program for area of a cylinder



Formula of surface area of cylinder:

Surface_area = 2 * Pie * r * (r + h)

Formula of volume of cylinder:

$\text{Volume} = \text{Pie} * r * r * h$

Pie = 22/7 or 3.14159265358979323846264338327950288419716939937510...

C code:

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

int main(){

    float r,h;
    float surface_area,volume;

    printf("Enter size of radius and height of a cylinder : ");
    scanf("%f%f",&r,&h);

    surface_area = 2 * M_PI * r * (r + h);
    volume = M_PI * r * r * h;

    printf("Surface area of cylinder is: %.3f",surface_area);
    printf("\nVolume of cylinder is : %.3f",volume);

    return 0;
}
```

Sample output:

```
Enter size of radius and height of a cylinder: 4 10
Surface area of cylinder is: 351.858
Volume of cylinder is: 502.655
```

PAPPU-14024

