Area of ellipse

```
#include<bits/stdc++.h>
using namespace std;

// Function to find area of an
// ellipse.
void findArea( float a, float b)
{
    float Area;

    // formula to find the area
    // of an Ellipse.
    Area = 3.142 * a * b ;

    // Display the result
    cout << "Area: " << Area;
}

// Driver code
int main()
{
    float a = 5, b = 4;

    findArea(a, b);

    return 0;
}
```

Area of triangle

```
#include <stdio.h>

int  main()
{
```

```
    float base, height, area;

    /* Input base and height of triangle */
    printf("Enter base of the triangle: ");
    scanf("%f", &base);
    printf("Enter height of the triangle: ");
    scanf("%f", &height);

    /* Calculate area of triangle */
    area = (base * height) / 2;

    /* Print the resultant area */
    printf("Area of the triangle = %.2f sq. units", area);

    return 0;
}
```

When 3 Sides are given

```
1.  #include<stdio.h>
2.  #include<math.h>
3.
4.  int main()
5.  {
6.      double a, b, c, s, area;
7.
8.      printf("Enter the sides of
  triangle\n");
9.
10.     scanf("%lf%lf%lf", &a, &b, &c);
11.
12.     s = (a+b+c)/2;
13.
```

```
14.     area = sqrt(s*(s-a)*(s-b)*(s-c));
15.
16.     printf("Area of the triangle =
  %.2lf\n", area);
17.
18.     return 0;
19. }
```

Volume of a cone

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

#define PI 3.14159

int main(){
    float radius, height, volume;
    printf("Enter base radius and height of a
Cone\n");
    scanf("%f %f", &radius, &height);

    /* Volume of Cone = 1/3 X PI X Radius X Radius
X Height */
    volume = 1.0/3 *(PI*radius*radius*height);

    printf("Volume of Cone : %0.4f\n", volume);

    getch();
    return 0;
}
```

Integer to Roman

```c
#include<stdio.h>

void decimal2roman(int num){
   int decimal[] = {1000,900,500,400,100,90,50,40,10,9,5,4,1}; //base values
   char *symbol[] = {"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};  //roman symbols
   int i = 0;

   while(num){ //repeat process until num is not 0
      while(num/decimal[i]){  //first base value that divides num is largest base value
         printf("%s",symbol[i]);    //print roman symbol equivalent to largest base value
         num -= decimal[i];  //subtract largest base value from num
      }
      i++;    //move to next base value to divide num
   }
}

int main()
{
   printf("250 -> ");
   decimal2roman(250);

   printf("\n1550 -> ");
   decimal2roman(1550);

   printf("\n670 -> ");
   decimal2roman(670);

   return 0;
}
```

Java program

```java
class Roman{
    public static String IntegerToRoman(int n){
        String roman="";
        int repeat;
        int magnitude[]={1000,900, 500, 400, 100, 90, 50,
40, 10, 9, 5, 4, 1};
        String symbol[]={"M","CM", "D", "CD", "C", "XC",
"L", "XL", "X", "IX", "V", "IV", "I"};

        for(int x=0; x<magnitude.length; x++){
            repeat=n/magnitude[x];
            for(int i=1; i<=repeat; i++){
```

```
                roman=roman + symbol[x];
            }
            n=n%magnitude[x];
        }
        return roman;
    }

    public static void main(String args[]){
        System.out.println("12: "+IntegerToRoman(12));
        System.out.println("999: "+IntegerToRoman(999));
    }
}
```

Print the reversed binary

```
public class
BinaryRotate
{
    public int rotateBinary(int number){
    int res = 0;
    while(number>0){
    res=res<<1;
    res = res|(number & 1);
    number=number>>1;
    }
    return res;
    }
    public static void main(String args[]){
    int x =30;
    BinaryRotate b = new BinaryRotate();
    System.out.println("Binary rotation of "+ x + " is : " + b.rotateBinary(x)
    }
    }
```

Arithmetic series

```
class class Main
{
static void printAP(int a, int d, int n)
{

    // Printing AP by simply adding d
    // to previous term.
    int curr_term;
curr_term=a;
    for (int i = 1; i <= n; i++)
    { System.out.print(curr_term + " ");
    curr_term =curr_term + d;

    }
}

// Driver code
public static void main(String[] args)
{
// starting number
int a = 2;

// Common difference
int d = 1;

// N th term to be find
int n = 5;

printAP(a, d, n);
}
}
```

Triangular sequence

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

// Returns true if 'num' is triangular, else false
int isTriangular(int num)
{
    // Base case
    if (num < 0)
        return 0;

    // A Triangular number must be sum of first n
    // natural numbers
    int sum = 0;
    for (int n=1; sum<=num; n++)
    {
        sum = sum + n;
        if (sum==num)
            return 1;
    }

    return 0;
}
// Driver code
int main()
{
    int n,a[n],i,flag=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        if(isTriangular(a[i]))
        {
            continue;
        }
        else
        {
            flag=1;
        }
    }
    if(flag==1)
        printf("Not a triangular sequence");
    else
         printf("Trinagular sequence");
}
```

Pig latin

```
class Main {
static boolean isVowel(char c) {
    return (c == 'A' || c == 'E' || c == 'I' || c == 'O'
|| c == 'U' ||
            c == 'a' || c == 'e' || c == 'i' || c == 'o'
|| c == 'u');
}

static String pigLatin(String s) {

    // the index of the first vowel is stored.
    int len = s.length();
    int index = -1;
    for (int i = 0; i < len; i++)
    {
        if (isVowel(s.charAt(i))) {
        index = i;
        break;
    }
    }

    // Pig Latin is possible only if vowels
    // is present
    if (index == -1)
        return "-1";

    // Take all characters after index (including
    // index). Append all characters which are before
    // index. Finally append "ay"
    return s.substring(index) +
           s.substring(0, index) + "ay";
}

// Driver code
public static void main(String[] args) {
```

```
        String str = pigLatin("graphic");
        if (str == "-1")
            System.out.print("No vowels found." +
                             "Pig Latin not possible");

        else
            System.out.print(str);
}
}
```

Comparing hexa and decimal

```
#include <stdio.h>
#include <math.h>
int main()
{
   char hex[17];
   long long decimal,dec, place;
   int i = 0, val, len;
   decimal = 0;
   place = 1;
   /* Input hexadecimal number from user */
   printf("Enter any hexadecimal number: ");
   gets(hex);
   printf("Enter the decimal number");
   scanf("%d",&dec);
   /* Find the length of total number of hex digit */
   len = strlen(hex);
   len--;
   /*
    * Iterate over each hex digit
    */
   for(i=0; hex[i]!='\0'; i++)
   {

      /* Find the decimal representation of hex[i] */
      if(hex[i]>='0' && hex[i]<='9')
      {
         val = hex[i] - 48;
      }
      else if(hex[i]>='a' && hex[i]<='f')
      {
```

```
            val = hex[i] - 97 + 10;
        }
        else if(hex[i]>='A' && hex[i]<='F')
        {
            val = hex[i] - 65 + 10;
        }
        else
        {
            printf("Invalid");
            return 0;
        }
        decimal += val * pow(16, len);
        len--;
    }

    if(decimal==dec)
    {

    printf("Equal");
    }
    else
    {
        printf("Not equal");
    }
    return 0;
}
```