**Experiment no 1:** WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or scalene. Take sides of the triangle as input from a user.

**Soultion:-**

**Code:-**

#include<stdio.h>

int main()

{

int x,y,z;

printf("enter the sides of triangles\n");

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

if((x==y)&&(y==z))

{

printf("equilateral");

}

else if((x==y)||(y==z)||(z==x))

{

printf("isosceles");

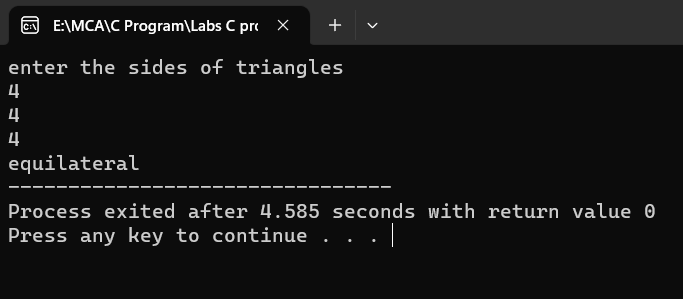
}

else

{

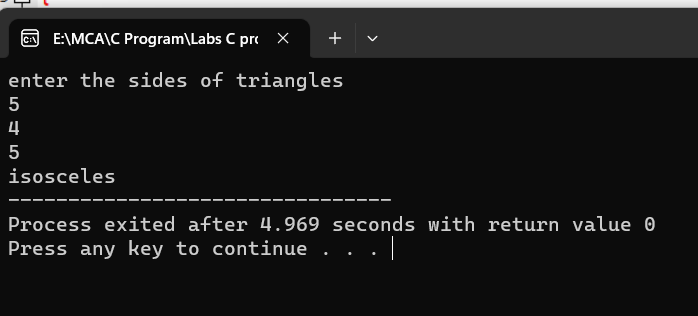
printf("scalen");

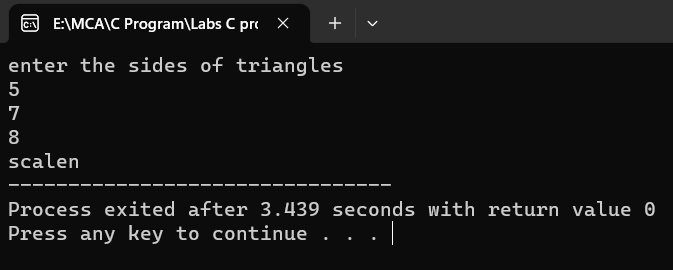
}

 return 0;

}

**Output:-**

****

****

**Experiment no 2:** WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI= weight(kgs)/Height(Mts)\*Height(Mts).

|  |  |
| --- | --- |
|  | **BMI** |
| **Starvation** | **<15** |
| **Anorexic** | **15.1 to 17.5** |
| **Underweight** | **17.6 to 18.5** |
| **Ideal** | **18.6 to 24.9** |
| **Overweight** | **25 to 25.9** |
| **Obese** | **30 to 39.9** |
| **Morbidity Obese** | **40.0 above** |

**Solution:-**

**Code:-**

#include <stdio.h>

int main()

{

float weight, height, bmi;

// Input weight in kilograms

printf("Enter your weight in kilograms: ");

scanf("%f", &weight);

// Input height in meters

printf("Enter your height in meters: ");

scanf("%f", &height);

// Calculate BMI

bmi = weight / (height \* height);

// Print the BMI value

printf("Your BMI is: %.1f\n", bmi);

// Determine the BMI category and print it

if (bmi < 15.0)

{

printf("BMI Category: Starvation\n");

}

else if (bmi >= 15.1 || bmi <= 17.5)

{

printf("BMI Category: Anorexic\n");

}

else if (bmi >= 17.6 || bmi <= 18.5)

{

printf("BMI Category: Underweight\n");

}

else if (bmi >= 18.6 || bmi <= 24.9)

{

printf("BMI Category: Ideal\n");

}

else if (bmi >= 25.0 || bmi <= 25.9)

{

printf("BMI Category: Overweight\n");

}

else if (bmi >= 30.0 || bmi <= 39.9)

{

printf("BMI Category: Obese\n");

}

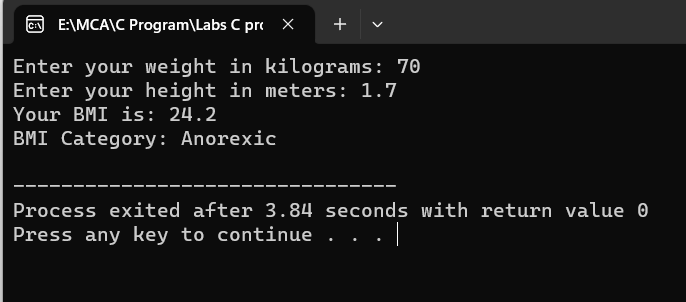
else

{

printf("BMI Category: Morbidity Obese\n");

}

return 0;

}

**Output:-**

**Experiment no 3:** WAP to check if three points (x1,y1), (x2,y2) and (x3,y3) are collinear or not.

**Solution:-**

**Code:-**

#include<stdio.h>

int main()

{ /\*collinear or not\*/

int x1,y1,x2,y2,x3,y3,p;

printf("enter the value of x1 and y1");

scanf("%d%d",&x1,&y1);

printf("enter the value of x2 and y2\n");

scanf("%d%d",&x2,&y2);

printf("enter the value of x3 and y3\n");

scanf("%d%d",&x3,&y3);

p=x1\*(y2-y3)+x2\*(y3-y1)+x3\*(y1-y2);

if(p==0)

{

printf("it is collinear");

}

else

{

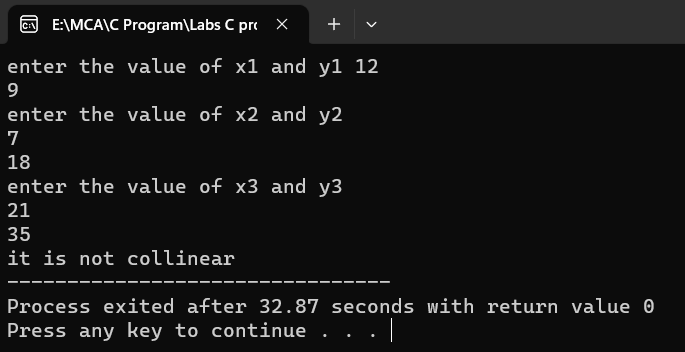
printf("it is not collinear");

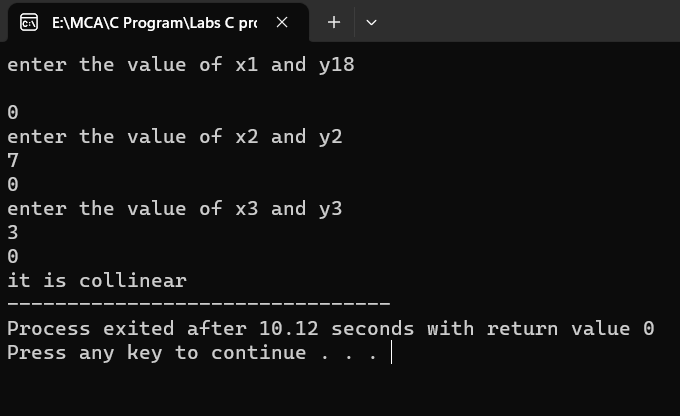
}

return 0;

}

**Output:-**

****

****

**Experiment No 4:** According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is input through the keyboard write a program to find out what is the day on 1st January of this year.

**Solution:-**

**Code:-**

#include<stdio.h>

#include<conio.h>

int main()

{

//basic year is chosen as the least year which user can enter through the keyboard where it should be Monday on 01th of January

int year, basic\_year=1900, leap\_year, remaining\_year, total\_days, day;

printf("Enter the year: ");

scanf("%d", &year);

year = (year-1)-basic\_year; //we are calculating the total years between basic year and input year

//Now calculate the leap years

leap\_year = year/4;

//calculate the year which are not leap years

remaining\_year = year - leap\_year;

//calculate total days present in all years

//remaining years are not leap year so total days in single year will be 365

//But in leap year we have 366 days in a single year

total\_days = (remaining\_year\*365) + (leap\_year\*366) + 1;

//Find the actual day

day = total\_days%7;

if(day==0)

printf("Monday");

else if(day==1)

printf("Tuesday");

else if(day==2)

printf("Wednesday");

else if(day==3)

printf("Thursday");

else if(day==4)

printf("Friday");

else if(day==5)

printf("Saturday");

else if(day==6)

printf("Sunday");

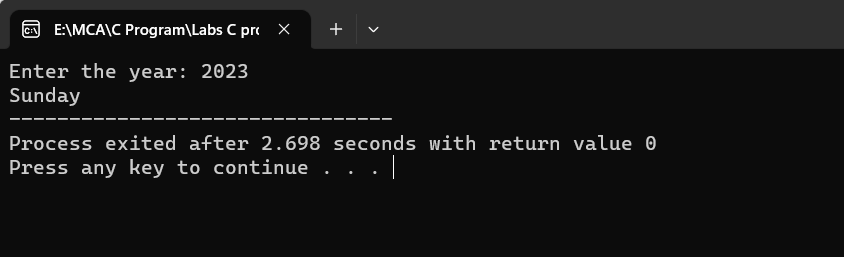
else

printf("Wrong Entry");

return 0;

}

**Output:-**

****

**Experiment 5:** WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three.

**Solution:-**

**Code:-**

#include<stdio.h>

int main()

{

/\*ternory opretor\*/

int a,b,c;

printf("enter value a");

scanf("%d",&a);

printf("enter value b");

scanf("%d",&b);

printf("enter value c");

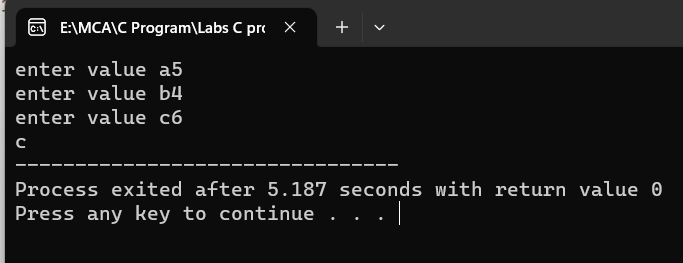
scanf("%d",&c);

a>b?((a>c)?printf("a"):printf("c")):((b>c)?printf("b"):printf("c"));

return 0;

}

**Output:-**

****