# **Assignment Question 1**

## **Documentation**

Two key things to check is to inform user if the triangle is valid and the type of triangle it is.

**Step 1.**

Prompt the User to key in the numbers for all 3 sides.

Printf(“Please enter a value\n”)

Scanf(%d, SideA)

**Step 2.**

System to verify if the triangle is valid

Algorithm need to use part A only.

a.       Check if Valid :

(a+b)>=c , (b+c)>=a, (c+a)>=b

eg. (1+2)>=3, (2+3)>=1, (3+1)>=2

{Valid = 1;}

b.       Example of Invalid :

(a+b)<=c, (b+c)<=a, (c+a)<=b

eg (1+1)<=3, (1+3)<=1, (3+1)<=1

{Valid = 0;}

**Step 3.**

Inform user if the triangle is valid or not

If (valid = 1);

{ Printf( “Triangle is valid \n”) ;}

Else

{ printf(“Triangle is invalid \n”);}

**Step 4.**

System to check if triangle is Equilateral, Scalene or Isosceles

a.       Equilateral means all 3 sides are of equal length.

Eg. side A, side B and side C are all same length

b.       Isosceles means at least 2 sides are of equal length.

Eg. side A and side B is the same length

c.       Scalene means all 3 sides are not of equal length.

Eg. side A, side B and side C are all difference length

**Step 5.**

Inform user the type of triangle it is.

Eg.: if (a==b, b==c, a==c);

{Printf(“Triangle is Equilateral”);

**Step 6.**

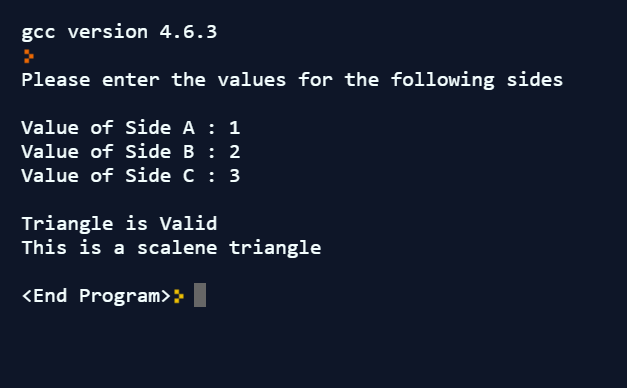
Indicate the end of program.

Printf(“<End of Program>”);

## **Test Result Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Inputs** | **Expected Outputs** | **Algorithm Outputs** | **Program Success/Failure** |
| Inform user if the triangle is valid and the type of triangle it is | User to key in the numbers for all 3 sides | Inform user if the triangle is valid or invalid and the type of triangle | (a+b)>=c , (b+c)>=a, (c+a)>=b | Success |
| Inform user the type of triangle | User to key in the numbers for all 3 sides | Inform user the type of triangle (Equilateral, Scalene or Isosceles) | (a==b) && (b==c) | Success |

## **Screenshot of the result**



# **Assignment Question 2**

## **Documentation**

The concept is to break down the value of the change into cents worth of coins. Eg. 5x 50 Cents.

In this case, user keys in value worth in cents between 5 -95 cents.

The breakdown of the cents should look like below.

Eg. 97 cents:

               1 x 50 = 50 Cents

               2 x 20 = 40 cents

               0 x 10 = 0 cents

               1 x 5 = 5 Cents

               2 x 1 = 2 cents

*Suggested method to code:*

**Step 1**

Prompt the User to key in the value.

Printf(“Please enter the value\n”)

Scanf(%d, change)

**Step 2**

System to read out the total change to be given.

This is only to cross check the value keyed in.

**Step 3**

Create the algorithm for the breakdown. Starting with 50cents to 5 cents.

**Step 4**

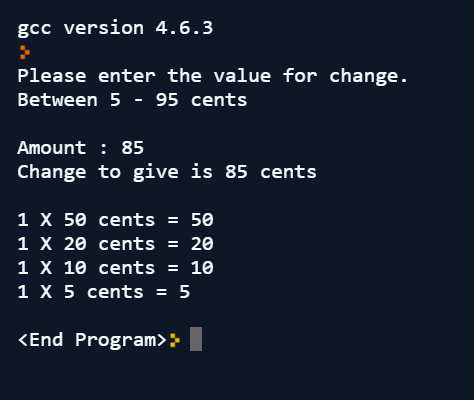
Indicate the end of program.

Printf(“<End of Program>”);

## **Test Result Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Inputs** | **Expected Outputs** | **Algorithm Outputs** | **Program Success/Failure** |
| Break down the value of the change to be given in coins | User to key in the amount | The breakdown of the coins from 50 cents to 5 cents | Fifty=change/50  Balance=change%50 | Success |

## **Screenshot of the result**



# **Assignment Question 3**

## **Documentation**

Similar to Q2, have to break down the value of the change into cents worth of coins. Eg. 5x 50 Cents.

In this case, user keys in value worth in dollars and cents unlike Q2 where only cents is requested.

Need to use similar algorithm as question 2. However, for this case, separate the combines dollar and cents into two portions.

First portion is the breakdown of the dollar.

This portion can use the algorithm similar to Q2.

Eg. $143 :

                1 x $100 =$100

                0 x $50 = $0

                2 x $20 = $40

                0 x $10 = $0

                0 x $5 = $0

                1 x $2 = $2

                1 x $1 = $1

Second portion is the breakdown of the cents.

This portion can use the algorithm directly from Q2.

*Suggested method to code:*

**Step 1**

Prompt the User to key in the value.

Printf(“Please enter the value\n”)

Scanf(%d, change)

**Step 2**

System to read out the total change to be given.

This is only to cross check the value keyed in.

**Step 3**

System to separate the dollar and the cents using ‘float or double’

**Step 4**

Follow the algorithm for based on Q2 for the two portions.

**Step 5**

Indicate the end of program.

Printf(“<End of Program>”);

## **Test Result Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Inputs** | **Expected Outputs** | **Algorithm Outputs** | **Program Success/Failure** |
| Break down the value of the change to be given in dollars and coins | User to key in the amount | The breakdown of the dollar from $100 to $1 and coins from 50 cents to 5 cents | Hundreddollar=change/100  Balance=change/100  Fiftycents=change/50  Balance=change%50 | Success |

## **Screenshot of the result**

