1. Write a C Program to Find the Smallest Number among Three Numbers (integer values) using Nested IF-Else statement.

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 90 -9 -8 | -9 is the smallest number. |
| Test Case 2 | 100 200 400 | 100 is the smallest number. |
| Test Case 3 | 20 30 10 | 10 is the smallest number. |
| Test Case 4 | 77 -44 99 | -44 is the smallest number. |

#include <stdio.h>

int main()

{

int n1, n2, n3;

scanf("%d %d %d", &n1, &n2, &n3); /\* where three number are read from the test cases and are stored in the variable n1, n2 and n3 \*/

/\* Complete the program to get the desired output \*/

/\* Only use the printf statement given below to exactly match your output

with the output cases. Change the variable n1 as required.

printf("%d is the smallest number.", n1); //Copy and paste this printf statement wherever required.\*/

if (n1<n2)

{

if(n1<n3)

printf("%d is the smallest number.", n1);

else

printf("%d is the smallest number.", n3);

}

else

{

if(n2<n3)

printf("%d is the smallest number.", n2);

else

printf("%d is the smallest number.", n3);

}

}

1. The length of three sides are taken as input. Write a C program to find whether a triangle can be formed or not. If not display “This Triangle is NOT possible.” If the triangle can be formed then check whether the triangle formed is equilateral, isosceles, scalene or a right-angled triangle. (If it is a right-angled triangle then only print Right-angle triangle do not print it as Scalene Triangle).

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 3 4 5 | Right-angle Triangle |
| Test Case 2 | 5 12 13 | Right-angle Triangle |
| Test Case 3 | 10 4 6 | Triangle is not possible |
| Test Case 4 | 7 6 8 | Scalene Triangle |
| Test Case 5 | 9 9 9 | Equilateral Triangle |

#include<stdio.h>

int main()

{

int a,b,c;

scanf("%d %d %d",&a, &b, &c); /\*The length of three sides are entered from the test cases \*/

/\* Complete the program. Copy and paste from the printf statements mentioned below wherever required for printing the outputs

printf("Triangle is not possible");

printf("Right-angle Triangle");

printf("Isosceles Triangle");

printf("Equilateral Triangle");

printf("Scalene Triangle");

\*/

if(a<(b+c)&&b<(a+c)&&c<(a+b))

{

if(a==b&&a==c&&b==c)

printf("Equilateral Triangle");

else if(a==b||a==c||b==c)

printf("Isosceles Triangle");

else

if((a\*a)==(b\*b)+(c\*c)||(b\*b)==(a\*a)+(c\*c)||(c\*c)==(a\*a)+(b\*b))

printf("Right-angle Triangle");

else if(a!=b&&a!=c&&b!=c)

printf("Scalene Triangle");

}

else

printf("Triangle is not possible");

}

1. Write a program to find the factorial of a given number using while loop.

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 7 | The Factorial of 7 is : 5040 |
| Test Case 2 | 11 | The Factorial of 11 is : 39916800 |
| Test Case 3 | 5 | The Factorial of 5 is : 120 |
| Test Case 4 | 10 | The Factorial of 10 is : 3628800 |

#include<stdio.h>

void main()

{

int n;

long int fact; /\* n is the number whose factorial we have to find and fact is the factorial \*/

scanf("%d",&n); /\* The value of n is taken from test cases \*/

/\* complete the program. Use the printf statements in the format mentioned below

to match your output exactly with output test cases

printf("The Factorial of %d is : %ld",n,fact);

You can declare any other variables if required \*/

int i=1;

fact = 1;

while(i<=n)

{

fact\*=i;

i++;

}

printf("The Factorial of %d is : %ld",n,fact);

}

1. Write a C program to find power of a number using while loops. The base number (>0) and exponent (>=0) is taken from the test cases.

Sample Test Cases

|  |  |  |
| --- | --- | --- |
|  | **Input** | **Output** |
| Test Case 1 | 6  0 | The result is : 1 |
| Test Case 2 | 4  8 | The result is : 65536 |
| Test Case 3 | 5  4 | The result is : 625 |
| Test Case 4 | 7  3 | The result is : 343 |

#include <stdio.h>

int main()

{

int base, exponent;

long int result;

scanf("%d", &base); //The base value is taken from test case

scanf("%d", &exponent); //The exponent value is taken from test case

if(exponent == 0)

result = 1;

else

{

result = 1;

while(exponent != 0)

{

result = result \* base;

--exponent;

}

}

printf("The result is : %ld\n", result);

return 0;

}