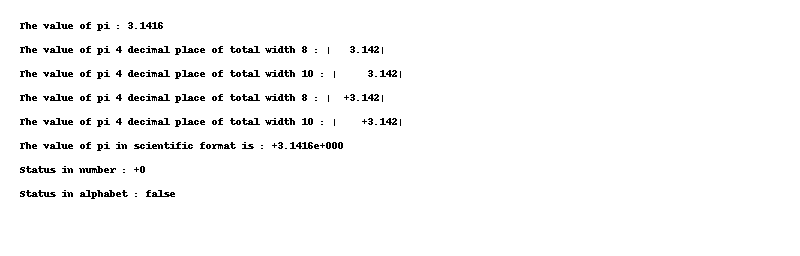
# Code: 1-1.cpp

/\*  
Write a program to display the formatted output as given below.  
Formatting the output :  
----------------------------  
The value of pi : 3.1416  
The value of pi 4 decimal place of total width 8 : | 3.1416|  
The value of pi 4 decimal place of total width 10 : | 3.1416|  
The value of pi 4 decimal place of total width 8 : |--3.1416|  
The value of pi 4 decimal place of total width 10 : |----3.1416|  
The value of pi in scientific format is : 3.1416e+00  
Status in number : 0  
Status in alphabet : false  
\*/  
  
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main() {  
 double pi = 3.1416;  
 cout << "The value of pi : " << pi << endl;  
 cout << "The value of pi 4 decimal place of total width 8 : |" << setw(8) << setprecision(4) << pi << "|" << endl;  
 cout << "The value of pi 4 decimal place of total width 10 : |" << setw(10) << setprecision(4) << pi << "|" << endl;  
 cout << "The value of pi 4 decimal place of total width 8 : |" << setw(8) << setprecision(4) << showpos << pi << "|" << endl;  
 cout << "The value of pi 4 decimal place of total width 10 : |" << setw(10) << setprecision(4) << showpos << pi << "|" << endl;  
 cout << "The value of pi in scientific format is : " << scientific << pi << endl;  
 cout << "Status in number : " << boolalpha << 0 << endl;  
 cout << "Status in alphabet : " << boolalpha << false << endl;  
 return 0;  
}

## Output



# Code: 1-10.cpp

/\*  
Formulate statements to perform the following:  
a. Left-justify the number 0.123456 in an output field with a width of 15.  
b. Output the number 23.987 as a fixed point number rounded to two decimal places,  
right-justifying the output in a field with a width of 12.  
c. Output the number â€“123.456 as an exponential and with four decimal spaces. How useful is  
a field width of 10?  
\*/  
  
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main() {  
 cout << left << setw(15) << 0.123456 << endl;  
 cout << right << setw(12) << fixed << setprecision(2) << 23.987 << endl;  
 cout << right << setw(10) << scientific << setprecision(4) << -123.456 << endl;  
 return 0;  
}

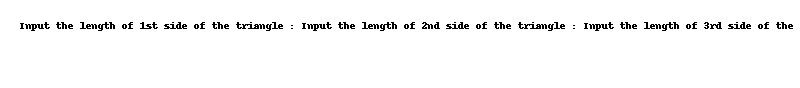
## Output



# Code: 1-2.cpp

/\*  
Write a C++ program to find the area of any triangle using Heron's formula.  
Visual Presentation:  
  
Find the area of any triangle using Heron's Formula :  
----------------------------------------------------------  
Input the length of 1st side of the triangle : 5  
Input the length of 2nd side of the triangle : 5  
Input the length of 3rd side of the triangle : 5  
The area of the triangle is : 10.8253\*/  
  
#include <iostream>  
#include <cmath>  
using namespace std;  
  
int main() {  
 double a, b, c, s, area;  
 cout << "Input the length of 1st side of the triangle : ";  
 cin >> a;  
 cout << "Input the length of 2nd side of the triangle : ";  
 cin >> b;  
 cout << "Input the length of 3rd side of the triangle : ";  
 cin >> c;  
 s = (a + b + c) / 2;  
 area = sqrt(s \* (s - a) \* (s - b) \* (s - c));  
 cout << "The area of the triangle is : " << area << endl;  
 return 0;  
}

## Output



# Code: 1-3.cpp

/\*  
Write a C++ program to input a single-digit number and print it in a rectangular form of 4 columns and  
6 rows.\*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
 int n;  
 cout << "Input a single-digit number: ";  
 cin >> n;  
 cout << n << n << n << n << endl;  
 cout << n << " " << " " << " " << n << endl;  
 cout << n << " " << " " << " " << n << endl;  
 cout << n << n << n << n << endl;  
 cout << n << " " << " " << " " << n << endl;  
 cout << n << " " << " " << " " << n << endl;  
 return 0;  
}

## Output



# Code: 1-4.cpp

/\*  
Write a C++ program that takes a number as input and prints its multiplication table up to 10.  
\*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
 int n;  
 cout << "Input a number: ";  
 cin >> n;  
 for (int i = 1; i <= 10; i++) {  
 cout << n << " \* " << i << " = " << n \* i << endl;  
 }  
 return 0;  
}

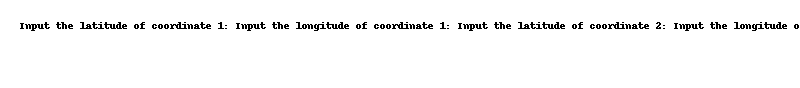
## Output



# Code: 1-5.cpp

/\*  
Write a C++ program to compute the distance between two points on the surface of the earth.  
\*/  
  
#include <iostream>  
#include <cmath>  
using namespace std;  
  
#define M\_PI 3.14159265358979323846  
  
int main() {  
 double x1, y1, x2, y2, d;  
 cout << "Input the latitude of coordinate 1: ";  
 cin >> x1;  
 cout << "Input the longitude of coordinate 1: ";  
 cin >> y1;  
 cout << "Input the latitude of coordinate 2: ";  
 cin >> x2;  
 cout << "Input the longitude of coordinate 2: ";  
 cin >> y2;  
 x1 = x1 \* M\_PI / 180;  
 y1 = y1 \* M\_PI / 180;  
 x2 = x2 \* M\_PI / 180;  
 y2 = y2 \* M\_PI / 180;  
 d = 6371.01 \* acos(sin(x1) \* sin(x2) + cos(x1) \* cos(x2) \* cos(y1 - y2));  
 cout << "The distance between the two points is: " << d << " km" << endl;  
 return 0;  
}

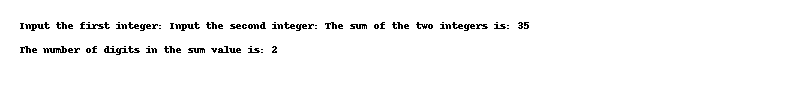
## Output



# Code: 1-6.cpp

/\*  
Write a C++ program to compute the sum of the two given integers and count the number of digits in  
the sum value.  
\*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
 int a, b, sum, count = 0;  
 cout << "Input the first integer: ";  
 cin >> a;  
 cout << "Input the second integer: ";  
 cin >> b;  
 sum = a + b;  
 cout << "The sum of the two integers is: " << sum << endl;  
 while (sum != 0) {  
 sum /= 10;  
 count++;  
 }  
 cout << "The number of digits in the sum value is: " << count << endl;  
 return 0;  
}

## Output



# Code: 1-7.cpp

/\*  
Input Format  
Input consists of the following space-separated values:  
int, long, char, float, and double, respectively.  
Output Format  
Print each element on a new line in the same order it was received as input.  
Note that the floating point value should be correct up to 3 decimal places  
and the double to 9 decimal places.  
Sample Input  
3 12345678912345 a 334.23 14049.30493  
Sample Output  
3 12345678912345 a 334.230 14049.304930000  
\*/  
  
#include <iostream>  
#include <iomanip>  
using namespace std;  
  
int main() {  
 int a;  
 long b;  
 char c;  
 float d;  
 double e;  
 cin >> a >> b >> c >> d >> e;  
 cout << a << endl;  
 cout << b << endl;  
 cout << c << endl;  
 cout << fixed << setprecision(3) << d << endl;  
 cout << fixed << setprecision(9) << e << endl;  
 return 0;  
}

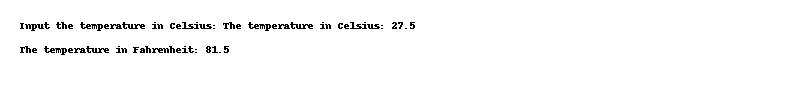
## Output



# Code: 1-8.cpp

/\*  
You can convert temperature from degrees Celsius to degrees Fahrenheit by multiplying by 9/5 and  
adding 32. Write a program that allows the user to enter a floating-point number representing degrees  
Celsius, and then displays the corresponding degrees Fahrenheit.  
\*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
 float celsius, fahrenheit;  
 cout << "Input the temperature in Celsius: ";  
 cin >> celsius;  
 fahrenheit = (celsius \* 9 / 5) + 32;  
 cout << "The temperature in Celsius: " << celsius << endl;  
 cout << "The temperature in Fahrenheit: " << fahrenheit << endl;  
 return 0;  
}

## Output



# Code: 1-9.cpp

/\*  
If you have two fractions, a/b and c/d, their sum can be obtained from the formula  
Write a program that encourages the user to enter two fractions, and then displays their sum in  
fractional form. (You donâ€™t need to reduce it to lowest terms.) The interaction with the user might look  
like this:  
Enter first fraction: 1/2  
Enter second fraction: 2/5  
Sum = 9/10  
You can take advantage of the fact that the extraction operator ( >> ) can be chained to read in  
more than one quantity at once:  
\*/  
  
#include <iostream>  
using namespace std;  
  
int main() {  
 int a, b, c, d, sum1, sum2;  
 cout << "Enter first fraction: ";  
 cin >> a;  
 cin.ignore();  
 cin >> b;  
 cout << "Enter second fraction: ";  
 cin >> c;  
 cin.ignore();  
 cin >> d;  
 sum1 = (a \* d) + (b \* c);  
 sum2 = b \* d;  
 cout << "Sum = " << sum1 << "/" << sum2 << endl;  
 return 0;  
}

## Output

