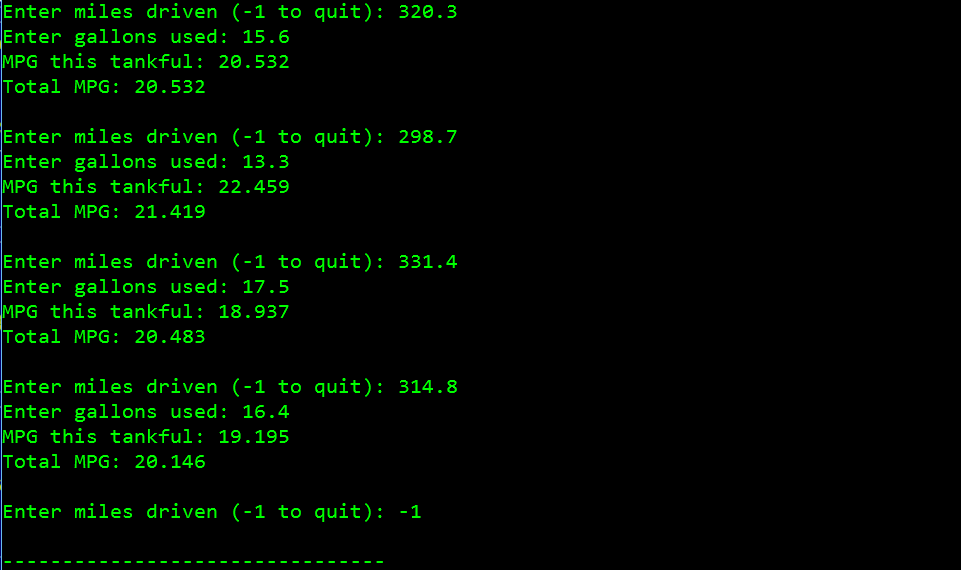
(Gas Mileage)

Some drivers are concerned with the mileage obtained by their automobiles. Write a program that keeps track of several tanks of gasoline by recording miles driven and gallons used for

each tank. Develop a C++ program that uses a while statement to input the miles driven and gallons used for each tankful. This while loop should terminate when the user enters the number of miles as -1. The program should calculate and display the miles per gallon obtained for each tank of gas and print the combined miles per gallon obtained for all tank up for each iteration. Make sure to set the width and precision as seen in the screenshot. See the example screenshot below for the expected output:



(Credit Card Limit Calculator) Write a C++ program that will determine whether a customer has exceeded the credit limit on a credit card account. For each customer, the following information is given:

a) Account number (an integer)

b) Balance at the beginning of the month

c) Total of all items charged by this customer this month

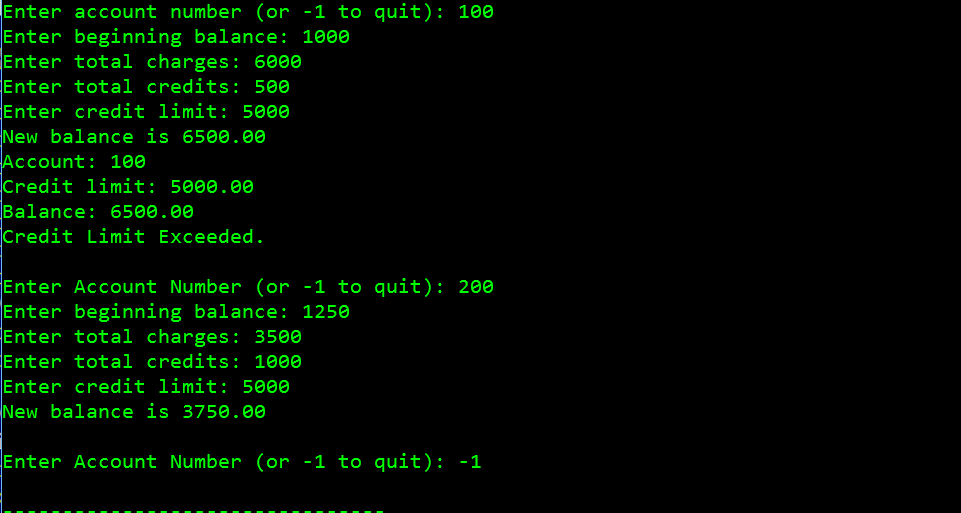
d) Total of all credits applied to this customer's account this month

e) Allowed credit limit

The program should use a while statement to input each of these pieces of information, calculate the new balance as the following:

new\_balance = beginning balance + charges – credits

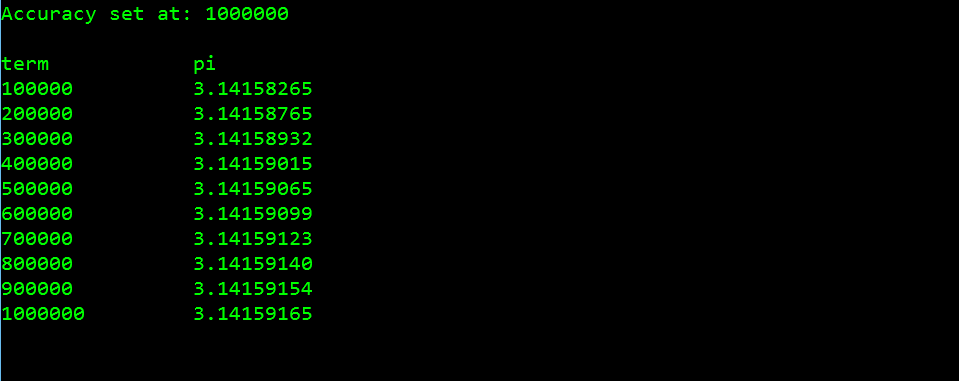
Determine whether the new balance exceeds the customer’s credit limit. For those customers whose credit limit is exceeded, the program should display the customer's account number, credit limit, new balance and the message “Credit Limit Exceeded.” This while loop should terminate when the user enters the account number of -1. See the example screenshot below for the expected output:



(Calculating π) Calculate the value of π from the infinite series: 

Print a table that shows the approximate value of π after each of the first 1,000,000 terms of this series.

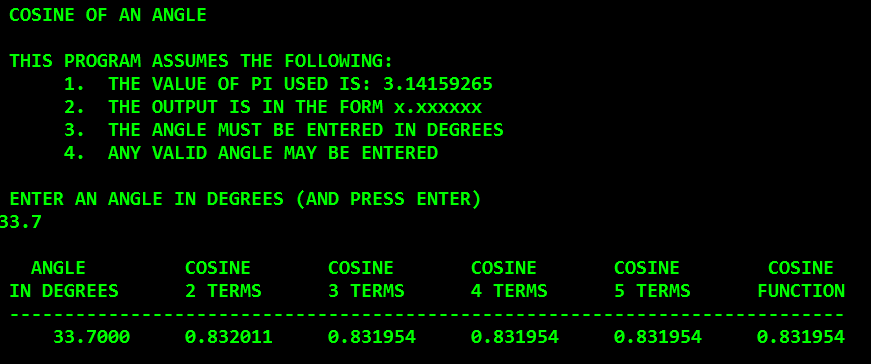
Have your program output the results in the following format:



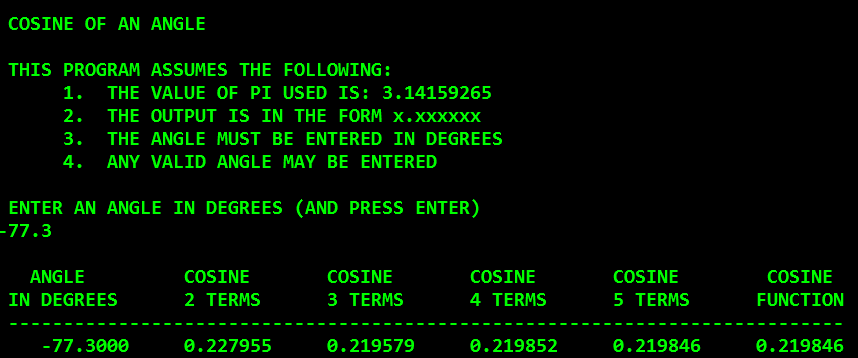
(Cosine of an Angle) Write a program that prompts the user to enter the degrees of an angle. Compute the cosine of the angle using the following series out to five terms ( the 8! term ):

The value of x is in radians, so to convert from degrees to radians, you have to multiply the degrees by π, then divide by 180.

Output the result of the cosine function found in the math library, to compare the accuracy of your program. See the example screenshots below for the expected output:



Another run of the program:



(Salesperson Salary Ranges) Use a one-dimensional array or vector to solve the following problem. A

company pays its salespeople on a commission basis. The salespeople each receive $200 per week

plus 9 percent of their gross sales for that week. For example, a salesperson who grosses $5000 in

sales in a week receives $200 plus 9 percent of $5000, or a total of $650. Write a program (using an

array of counters) that determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson’s salary is truncated to an integer amount):

a) $200–299

b) $300–399

c) $400–499

d) $500–599

e) $600–699

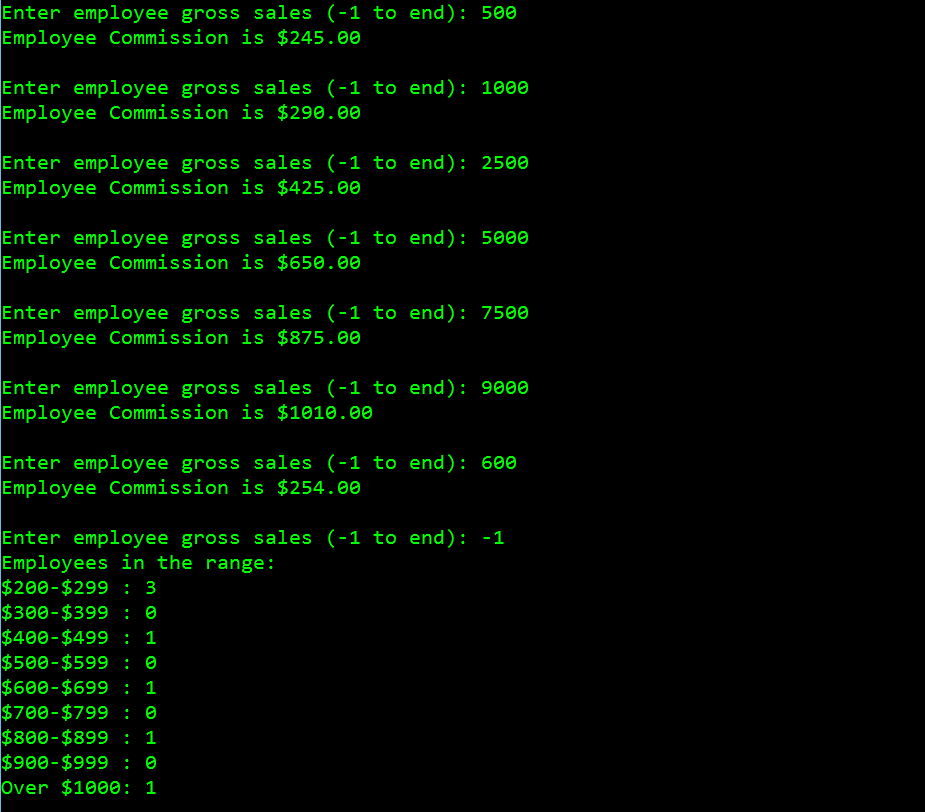
f) $700–799

g) $800–899

h) $900–999

i) $1000 and over

See the example screenshot below for the expected output:



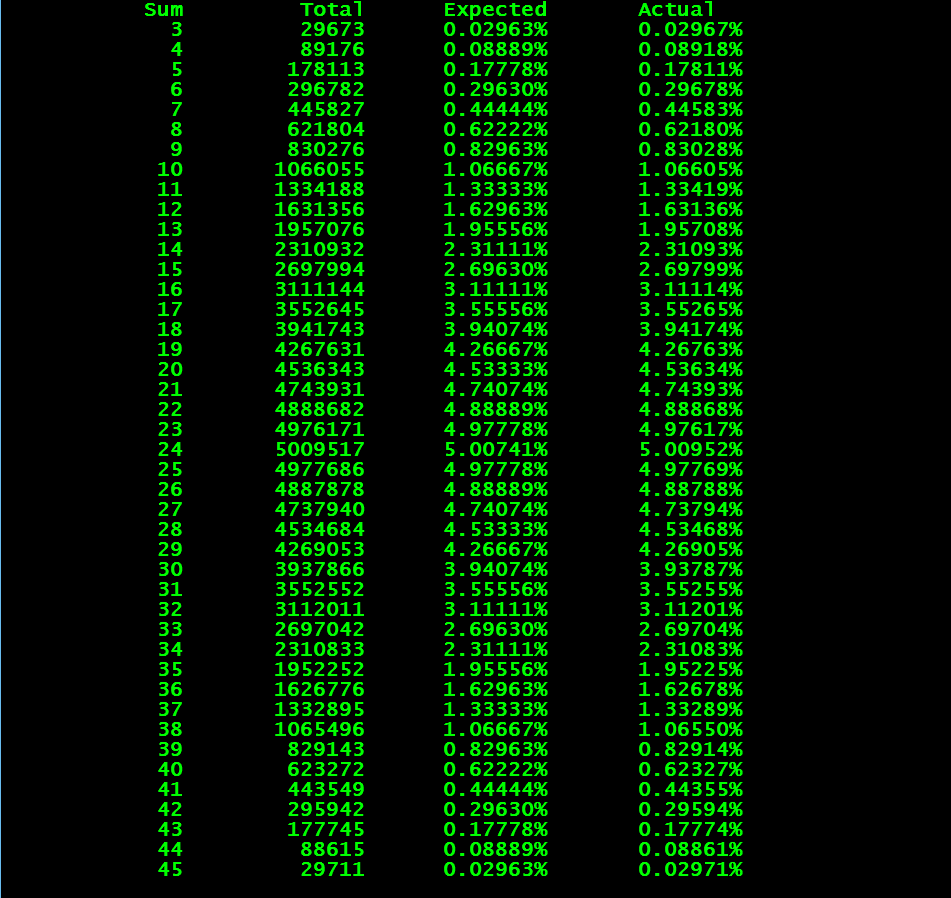
(Dice Rolling)

Suppose we are developing a role playing board game in which there are three 15 sided dice used in the game. Write a program that simulates the rolling of three dice. The program should use the random function to roll the first die and should use random again to roll the second die and third die.

The sum of the three values should then be calculated. Each die can show an integer value from 1 to 15, so the sum

of the three values will vary from 3 to 45.

Your program should roll the three dice 100,000,000 times. Use a one-dimensional array or vector to tally the numbers of times each possible sum appears. Print the results in a tabular format. See the example screenshot below for the expected output:



(Rounding Numbers) The floor function can be used to round a number to a specific decimal place. The following statement:

y = floor( x \* 10+ .5 ) / 10;

rounds the value of x (assuming x is of the data type double) to the tenths position (the first position to the right of the decimal point). Likewise, the statement:

y = floor( x \* 100+.5) / 100;

rounds x to the hundredths position (the second position to the right of the decimal point). Write a program that defines four functions to round a number x (of the type double) in various ways that returns a double:

a) roundToInteger( number )

b) roundToTenths( number )

c) roundToHundredths( number )

d) roundToThousandths( number )

For each value read, your program should print the original value, the number rounded to the nearest integer, the number rounded to the nearest tenth, the number rounded to the nearest hundredth and the number rounded to the nearest thousandth. See the example screenshot below for the expected output:

