

For Loops Challenge 11: Binary Hexadecimal Converter App

Description:

You are responsible for writing a program that will generate binary and hexadecimal values from 1 up to a specified user value. Recall that decimal is a base 10 number system, binary is a base 2 number system, and hexadecimal is a base 16 number system. Your program will use list slicing to first only show a portion of these values. Your program will then loop through the entire lists of decimal, binary, and hexadecimal values to show the relationship between numbers of different bases.

Step By Step Guide:

- Print a welcome message.
- Get user input for how many values they would like to convert to binary and hexadecimal.
- Using the range function, generate a list of numbers holding the decimal values from 1 up to the users maximum value.
- Create a blank list for the binary values.
- Create a blank list for the hexadecimal values.
- Use a for loop to loop through the decimal values. During each iteration:
 - Determine the binary representation and hexadecimal representation of the decimal value and add each value to the appropriate list.
 - To accomplish this use the built in bin() and hex() functions.
 - Google or check the python documentation on how to use these functions.
- Print a message informing the user that the lists are complete.
- Rather than print the whole list initially, use slicing to only show a portion of each list.
 - Get user input for the decimal number to start and stop at.
 - Be careful and think as to how these numbers relate to the indices of a list slice.
- Print a message for each list slice.
- Use a for loop to loop through the portion of the list specified and print each element.
- Prompt the user to press Enter to see the entire list generated.
 - To pause a program you can use an input statement.
- Print a table header.
- Using only one for loop, print the decimal, binary, and hexadecimal values for each element in each list.
 - This can be accomplished using the zip() function or proper list indexing.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Binary/Hexadecimal Converter App

Compute binary and hexadecimal values up to the following decimal number: 12

Generating lists....complete!

Using slices, we will now show a portion of each list.

What decimal number would you like to start at: 4

What decimal number would you like to stop at: 7

Decimal values from 4 to 7:

4
5
6
7

Binary values from 4 to 7:

0b100
0b101
0b110
0b111

Hexadecimal values from 4 to 7:

0x4
0x5
0x6
0x7

Press Enter to see all values from 1 to 12.

Decimal---Binary---Hexadecimal

1---0b1---0x1
2---0b10---0x2
3---0b11---0x3
4---0b100---0x4
5---0b101---0x5
6---0b110---0x6
7---0b111---0x7
8---0b1000---0x8
9---0b1001---0x9

10----0b1010----0xa

11----0b1011----0xb

12----0b1100----0xc

For Loops Challenge 12: Quadratic Equation Solver App

Description:

You are responsible for writing a program that will display the solutions to any number of quadratic equations. Your program will ask the user how many quadratic equations they would like to solve, ask for the coefficients of the equation in the standard form of $ax^2 + bx + c = 0$, solve for x, and then display the solutions. Your program will allow for both real and complex solutions.

Step By Step Guide:

- Print a welcome summary to the user.
 - This summary should describe a quadratic equation and complex numbers.
- Get user input for how many quadratic equations they would like to solve.
- Loop through the number of equations. Each iteration, you should:
 - Print a message header stating the equation number you are solving.
 - Get user input for the values of the coefficients a, b, and c.
 - Solve for the roots of the quadratic x1 and x2.
 - In order to solve a quadratic equation, you may be required to take the square root of a negative value which would result in an imaginary number. The resulting solution is a complex number as it has both real and imaginary parts.
 - The previously introduced math library's `sqrt()` function works well for real numbers but not for imaginary numbers. To work with imaginary values and complex numbers we will need to import a library of extra code.
 - Type `import cmath` as the first line of code in your program.
 - Print a summary of the solutions to the equation.
- Once the loop is complete, print a message thanking the user for using the program.
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Quadratic Equation Solver App.

A quadratic equation is of the form $ax^2 + bx + c = 0$

Your solutions can be real or complex numbers.

A complex number has two parts: $a + bj$

Where a is the real portion and bj is the imaginary portion.

How many equations would you like to solve today: 2

Solving equation #1

Please enter your value of a (coefficient of x^2): 1

Please enter your value of b (coefficient of x): 6

Please enter your value of c (coefficient): 9

The solutions to $1.0x^2 + 6.0x + 9.0 = 0$ are:

$$x_1 = (-3+0j)$$

$$x_2 = (-3+0j)$$

Solving equation #2

Please enter your value of a (coefficient of x^2): 1

Please enter your value of b (coefficient of x): -5

Please enter your value of c (coefficient): 14.2

The solutions to $1.0x^2 + -5.0x + 14.2 = 0$ are:

$$x_1 = (2.5+2.819574435974337j)$$

$$x_2 = (2.5-2.819574435974337j)$$

Thank you for using the Quadratic Equation Solver App. Goodbye.

For Loops Challenge 13: Factorial Calculator App

Description:

You are responsible for writing a program that will calculate the factorial of any given number. Your program will display the mathematical relationship of the factorial. It will then use the math library to compute the value of the given factorial. Lastly, your program will use its own algorithm to compute the value of the given factorial and compare the results.

Step By Step Guide:

- Print a welcome message.
- Get user input for a number to compute the factorial of.
- Print a mathematical representation of the factorial you are about to compute.
 - To accomplish this you will want to use a single print statement that prints the number chosen and " $! =$ ".
 - Then you will want to use the optional argument `end=" "` to override the default `\n` end of a print statement.
 - Setting `end=" "` will end the print statement with a space and keep the prompt on the same line.
 - Next, use a for loop to print through the sequence of numbers needed for your factorial and use an appropriate value for the end argument.
 - Lastly, outside of the for loop print your last number with the default end argument of `\n` to put your prompt on a new line.
- Use the math library to compute the factorial of your given number and print the result.
 - Type `import math` as the first line of code in your program.
 - The math library has a built in function to compute a factorial you can use.
- Write your own algorithm to compute the factorial of your given number and print the results.
 - Recall that a factorial is just repeated multiplication.
 - To perform this repeated multiplication, use a for loop with a numerical range.
- Print a summary of the results to show that math library and your algorithm were in agreement.
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Factorial Calculator App

What number would you like to compute the factorial of? 10

$10! = 1*2*3*4*5*6*7*8*9*10$

Here is the result from the math library:

The factorial of 10 is 3628800!

Here is the result from my own algorithm:

The factorial of 10 is 3628800!

It is shown twice that $10! = 3628800$ (with excitement)

For Loops Challenge 14: Fibonacci Calculator App

Description:

You are responsible for writing a program that will compute the first n terms of the Fibonacci Sequence. Your program will then display these terms. Next, your program will calculate the ratios of consecutive Fibonacci numbers to prove that these ratios approach the irrational mathematical constant of Phi; 1.618....

Step By Step Guide:

- Print a welcome message.
- Get user input for how many digits of the Fibonacci Sequence they would like to compute.
- Define a list which holds two integers, 1 and 1.
 - This is the start of your Fibonacci Sequence.
- Using a for loop, write an algorithm that will compute the next term of the Fibonacci Sequence and append it to your list.
 - The nth term of the Fibonacci Sequence can be found by adding the n-1 and n-2 terms together.
 - You essentially take the previous two terms in the sequence and add them together to get the current term.
- Iterate over this loop until you reach the desired number of terms.
- Display the results of your algorithm.
- Define an empty list that will hold the ratios of consecutive terms from your Fibonacci Sequence.
 - These ratios will approach a value of Phi; 1.618....
- Using a for loop, write an algorithm that will compute these ratios.
 - To compute this ratio, you will divide the 2nd term in your list by the first.
 - The next iteration, you will divide the 3rd term by the 2nd.
 - The next iteration, the 4th term by the 3rd, and so forth.
- Each time you compute this ratio, append it to your list.
- Display the results of your algorithm.
- Print a statement about the Golden Ratio and Fibonacci.
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Fibonacci Calculator App

How many digits of the Fibonacci Sequence would you like to compute: 15

The first 15 numbers of the Fibonacci sequence are:

1
1
2
3
5
8
13
21
34
55
89
144
233
377
610

The corresponding Golden Ratio values are:

1.0
2.0
1.5
1.6666666666666667
1.6
1.625
1.6153846153846154
1.619047619047619
1.6176470588235294
1.61818181818182
1.6179775280898876
1.6180555555555556
1.6180257510729614
1.6180371352785146

The ratio of consecutive Fibonacci terms approaches Phi; 1.618...

For Loops Challenge 15:

Grade Point Average Calculator App

Description:

You are responsible for writing a program that will collect any number of grades from a user. Your program will sort these grades numerically from highest to lowest and calculate the grade point average of the user. Your program will then ask for the average the user desires and calculate what the user must get on their next assignment to achieve this average. Lastly, your program will make a copy of the users grades and allow them to alter one of their previous grades to see how doing worse or better on an assignment would have changed their overall average.

Step By Step Guide:

- Print a welcome message.
- Get user input for their name.
- Get user input for the number of grades they would like to enter.
- Create an empty list that will hold the user's grades.
- Get user input for each individual grade and add it to the list.
- Sort the grades from highest to lowest and print each grade individually as formatted below.
- Calculate the average grade for the student as a float.
- Round this average to two decimal places.
- Display the students grade summary as formatted below.
 - This summary should include the user's name, the total number of grades, the highest grade, the lowest grade, and the average.
- Prompt the user for their desired average.
- Calculate what grade they need to get on their next assignment to achieve this average.
- Round this grade to two decimal places.
- Wish the user luck and inform them of the needed grade to get their desired average.
- Make a copy of your original list of grades.
- Get user input for a grade value that they would like to change.
- Get user input for the new grade value they would like the old grade changed to.
- Make the appropriate changes in the copy of the original list.
 - Please keep the original grades list in tact.
- Sort the new grades from highest to lowest and print each grade individually as formatted below.
- Calculate the new average grade for the student as a float.

- Round this average to two decimal places.
- Display the students grade summary as formatted below.
 - This summary should include the user's name, the total number of grades, the highest grade, the lowest grade, and the average.
- Print a comparison of the users old average and updated average.
- Print a statement regarding how many points, rounded to two decimal places, their average changed by.
- Inform them that their original grades however are still the same and print the original grades as a list.
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Average Calculator App

What is your name: mike

How many grades would you like to enter: 5

Enter grade: 72

Enter grade: 64

Enter grade: 80

Enter grade: 83

Enter grade: 77

Grades Highest to Lowest:

83

80

77

72

64

Mike's Grade Summary:

Total Number of Grades: 5

Highest Grade: 83

Lowest Grade: 64

Average: 75.2

What is your desired average: 79

Good luck Mike!

You will need to get a 98.0 on your next assignment to earn a 79.0 average.

Lets see what your average could have been if you did better/worse on an assignment.

What grade would you like to change: 64

What grade would you like to change 64 to: 80

New Grades Highest to Lowest:

83

80

80

77

72

Mike's New Grade Summary:

Total Number of Grades: 5

Highest Grade: 83

Lowest Grade: 72

Average: 78.4

Your new average would be a 78.4 compared to your real average of 75.2!

That is a change of 3.2 points!

Too bad your original grades are still the same!

[83, 80, 77, 72, 64]

You should go ask for extra credit!