Self-Review Questions
Functions

Note: Feel free to use extra functions to further break down your program. Sometimes that little bit of extra effort can make a program much smaller.

Q5. Write a function countCurrency() that takes an amount as a parameter and finds the minimum number of notes of different denominations that sum up to the given amount. Starting from the highest denomination note, try to accommodate as many notes as possible for a given amount. The function should print the amount of each note on a separate line.

We may assume we are limited to the following notes [5000, 1000, 500, 100, 20, 10, 1] (We have an infinite amount of each note).

In the main program, take `amount` as input and call the function.

Sample Input:

```
12345
```

Sample Output:

```
5000 : 2
1000 : 2
500 : 0
100 : 3
20 : 2
10 : 0
1 : 5
```

Q6. A shoe store is offering a sale to its clients on the purchase of two or more pairs of shoes. The first pair gets a discount of *10%*, the second pair gets a discount of *20%*, and each subsequent pair gets a discount of *30%*. Your task is to write a function to facilitate the billing process.

Write a function `bulk_price` that takes two parameters, `list_price` of type float and `items` of type int, that returns the wholesale price for all the shoes purchased in one order.

In the main program, take `list_price` and `items` as input on separate lines and call the function inside a print statement.

Note: Items >= 2

Sample Input:

```
100.00
10
```

Sample Output:

```
730.0
```

Q7. Write a function named `print_ladder` that takes one parameter of `size` of type int, and prints a ladder of asterisks (*) with a `size` number of steps and a gap of `size - 2` between steps. A step is a row of asterisks of length `size` separated by spaces. The width of the ladder should be consistent.

In the main program, take `size` as input and call the function.

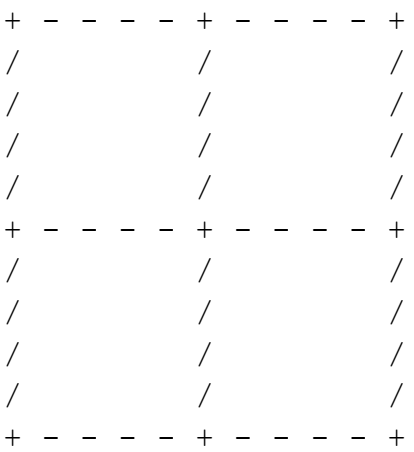
Sample Input:

```
4
```

Sample Output:

```
*       *
*       *
* * * *
*       *
*       *
* * * *
*       *
*       *
* * * *
*       *
*       *
* * * *
*       *
*       *
```

Q8. Generalizing a problem refers to changing hard coded values to a parameter. Consider the following grid.



Write a generalized function called `general_grid` that takes the following 5 parameters and draws a corresponding grid.
- `plus` is a string which will be used wherever '+' is used in the above grid
- `minus` is a string which will be used wherever '-' is used in the above grid
- `slash` is a string which will be used wherever '/' is used in the above grid
- `height` is the number of consecutive lines beginning with slash
- `width` is the number of consecutive occurrences of minus between two occurrences of plus . Each minus is surrounded on each side by a space.

The above grid would therefore be printed by
`general_grid('+', '-', '/', 4, 4)`

In the main program, take `plus`, `minus`, `slash`, `height` and `width` as input on separate lines and call the function.

Sample Input:

*

+

.

3

1

Sample Output:

```
* + * + *

.   .   .

.   .   .

.   .   .
* + * + *

.   .   .

.   .   .

.   .   .
* + * + *
```