LAB 6: FUNCTIONS – PART 1

For each problem below:

a) Analyze the problem by identifying input, output, formula, and constraint
b) Design an algorithm to solve the problem using pseudocode (so that you could include the pseudocode in your program)
c) Prepare several, appropriate number of test data to verify the correctness of your program
d) Prepare, compile, link, and execute the program to solve the problem
e) Test your program using the prepared test data
f) Write proper documentation in the program. Include the following information to form a banner at the beginning of your program:

```
/*******************************************************
* Author's: your name and student ID                   *
* Course: the course code only                         *
* Section: your specific section number                *
* Date: of lab session                                 *
* Brief description:    of what problem the            *
*                       program tries to solve         *
* Pseudocode: write the algorithm to solve the         *
*             problem                                  *
* Test data: provide a set of test data                *
*             - input & expected output                *
*******************************************************/
```

QUESTIONS

1. Write a program with multiple programmer-defined functions that displays the face of a smiling person as shown in Fig. 1. The structure chart in Fig. 2 shows how the main task can be broken apart. Different functions must be defined to perform different tasks.

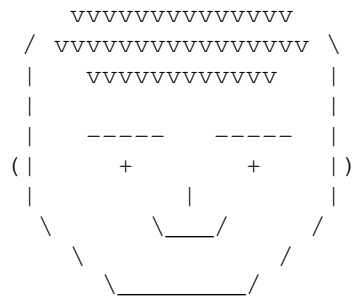


Fig. 1

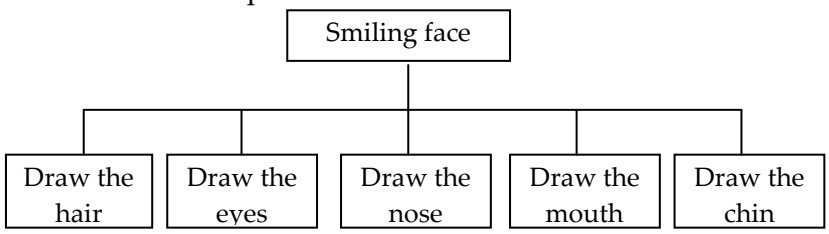
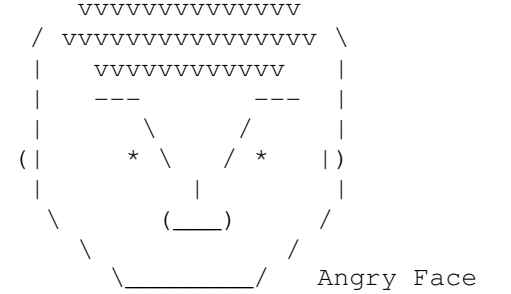
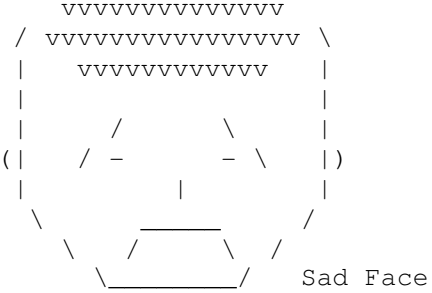


Fig. 2

2. Improve the program written for Question 1 so that along with the smiling face, the following faces may be displayed on request. Since the shape of the hair, the nose, and the chin is common to all faces, use the same functions as written in the earlier program.



Angry Face



Sad Face

You may use the following menu to request which one to display:

```
Do you wish:
      H - the Happy face
      S - the Sad face
      A - the Angry face
      Q - to Quit
Enter the choice >
```

3. Write a program to draw a multiple-level house based on chosen number of levels. The program may be organized into several functions as follows:
   - main() – prompt and ask user to enter the chosen number of levels and call the other functions appropriately to draw the house. For example, if the user asked for 3-level house, then call the function `drawUFloor()` twice.
   - drawRoof() – draw the roof as shown in Fig 3.
   - drawGFloor() – draw the ground floor as shown in Fig 4.
   - drawUFloor() – draw the upper floor as shown in Fig 5.

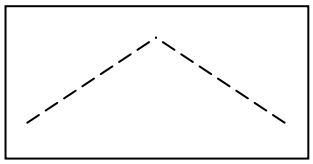


Fig. 3

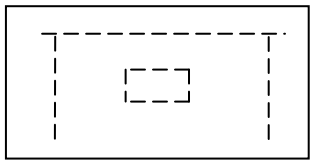


Fig. 4

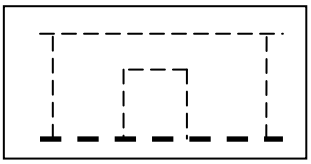


Fig. 5

4. Write a program that requires user to input two real numbers and one of the letter codes A, S, M, D or L. Based on the code entered by the user, the program needs to perform an appropriate arithmetic operation to the two numbers. Do all of these tasks in a function called `simpleCalculator`(). However, return the result of the operation to the `main()` function so that it can be printed. The letter codes and their operations are as listed below:

Letter Code	Operation
A	Addition
S	Subtraction
M	Multiplication
D	Division
L	Modulus

5. Improve the program written for Question (4) so that each of the arithmetic operations will be defined within individual programmer-defined function as listed in the table below:

Letter Code	Operation	Programmer-defined Function
A	Addition	`addNumbers()`
S	Subtraction	`subtractNumbers()`
M	Multiplication	`nultiplyNumbers()`
D	Division	`divideNumbers()`
L	Modulus	`modNumbers()`

The `main()` function prompts and gets the user's input for two numbers and a letter code. If the letter code is 'A', then calls the function `addNumbers()` and pass the two numbers read to the called function. Within the function `addNumbers()`, perform the addition operation onto both numbers and print the result of the arithmetic operation on the screen. Apply similar algorithm to the other letter codes.

6. Re-write the program written for Question (5) so that the result of the arithmetic operation is returned to the `main()` function. Then only the returned result will be displayed on screen.

7. Improve the calculator program written for Question (6) so that it is repeated *n*-times. Ask the user to enter the *n*-value. Also, include proper data validation to the program.