Title	Assignment 6
Due	20-Apr-2012 17:00
Number of resubmissions allowed	0
Grade	**78.0 (max 100.0)**
Modified by instructor	10-Apr-2012 22:10

Instructions

This tutorial is about functions and arrays.

Question 1

Write a Python module that contains the following 3 functions to create hollow boxes of stars.

- *print_square ()*, that prints a 5x5 box on the screen
- *print_rectangle (width, height)*, that prints a box on the screen with given width and height
- *get_rectangle (width, height)*, that returns a string containing a box with given width and height

A python module is a file containing Python code just like a regular program. Typically such a module will contain only functions. To use the module in another program, we first issue the *import* statement, then refer to each function by prefixing it with the module name.

Sample I/O:

```
Choose test:
a
*****
*   *
*   *
*   *
*****
```

Sample I/O:

```
Choose test:
b 3 4
calling function
***
* *
* *
***
called function
```

Sample I/O:

```
Choose test:
```

```
c 4 3
calling function
called function
****
*  *
****
```

Save your module as **boxes.py**. The main program has been supplied as **drawings.py** - use this to test your program and do not change this file.

Question 2

Mathematical functions map naturally to program functions and modules often are used to group such functions for reuse.

In the Gumatj* language, numbers use only the digits 0-4, such that instead of "tens", the second digit represents multiples of 5. Write a Python module with the following functions for simple Gumatj arithmetic, assuming that all values have at most 2 digits. *(Reference: http://en.wikipedia.org/wiki/Quinary)*

- *gumatj_to_decimal (a)*, that converts a Gumatj number to decimal
- *decimal_to_gumatj (a)*, that converts a decimal number to Gumatj
- *gumatj_add (a, b)*, that adds 2 Gumatj numbers
- *gumatj_multiply (a, b)*, that multiples 2 Gumatj numbers

Sample I/O:

```
Choose test:
d 12
calling function
called function
22
```

Sample I/O:

```
Choose test:
g 22
calling function
called function
12
```

Sample I/O:

```
Choose test:
a 12 14
calling function
called function
31
```

Sample I/O:

```
Choose test:
m 3 4
calling function
called function
22
```

Save your module as **gumatj.py**. The main program has been supplied as **base5.py** - use this to test your program and do not change this file.

Question 3

Programs are often refactored or reorganised to make better use of functions and thereby reduce redundancy and improve reusability of code.

You have been provided with the complete **combine.py** Python program that calculates the number of k-permutations of n items. A much more readable and space-efficient program is also provided as **fcombine.py**. You must reuse/adapt the code in the complete program to create the **mymath.py** module with the 2 required functions.

Sample I/O:

```
Enter n:
4
Enter k:
2
Number of permutations: 12
```

Save your module as **mymath.py**. The main program has been supplied as **fcombine.py** - use this to test your program and do not change this file.

Question 4

Write a program that takes in a list of marks (separated by spaces) and outputs a histogram representation of the marks according to the mark categories at UCT:

- fail < 50%
- 50% <= 3rd < 60%
- 60% <= lower 2nd < 70%
- 70% <= upper 2nd < 75%
- 1st >= 75%

From the given marks, count the number of marks that satisfy the conditions of each category. For example, for the following list of marks: 32 67 54 90 77, there is one fail, one 3rd, one lower 2nd, no upper 2nds and two firsts. Then print out horizontal bars, using "X", that correspond to the counters, along with some hard-coded axes and labels to represent a histogram.

You should use an array/list to store the counters.

Sample I/O:

```
Enter a space-separated list of marks:
12 23 34 45 56 67 78 89 90
1 |XXX
2+|
2-|X
3 |X
F |XXXX
```

Save your program as **histogram.py**.

Mark Weighting

Question 1 : 30
Question 2 : 20
Question 3 : 20
Question 4 : 30

Additional resources for assignment

- drawings.py (1 KB; 10-Apr-2012 19:48)
- base5.py (1 KB; 10-Apr-2012 19:48)
- fcombine.py (1 KB; 10-Apr-2012 19:48)
- combine.py (1 KB; 10-Apr-2012 22:09)

Submission

This assignment does not accept online submissions. Contact your instructor for additional instructions.

Done