**Programming Challenge Problems**

A graduated set of set of problems, all within the Python programming taught on the course.

## 1 Python Does Sums

***Knowledge required****: input and print; conversion to integers. No conditional or loops. After session 1.*

Write a program allowing the user to enter two numbers and output the product (one multiplied by the other). Here are two examples of the program running:

```
***** PYTHON CAN MULTIPLY *****
            TEST ME !!!
Enter x? 193
Enter y? 789
You asked me to multiply 193 by 789  1
93 * 789 = 152277

***** PYTHON CAN MULTIPLY *****
            TEST ME !!!
Enter x? 77
Enter y? 1293
You asked me to multiply 77 by 1293
77 * 1293 = 99561
```

## 2 Evil Dictator

***Knowledge required****: input and print; if statement. Loops can be used though there is a way to avoid them. After session 3 or 4.*

You are captured by an evil dictator who demands your name. Unknown to you, he cannot abide names containing an 'e' (or an 'E'). He has a program to check the names of his captives. Here is a transcript of his program running for you and your fellow captive Bill.

```
The evil dictator has captured you and demands your name.
Obey at once!
Enter your name, slave: Bill
Welcome Bill your name is acceptable
You can be my honoured slave!

The evil dictator has captured you and demands your name.
Obey at once!
Enter your name, slave: Ben
Oh no! A name with an 'e'
Guards, send Ben to the dungeon
```

To save yourself, offer to rewrite his program in the language Python, so far unknown is his evil kingdom.

## 3 Simple Calculator

***Knowledge required**: input and print; if statement and loop. After session 4.*

This program makes Python into a simple calculator, able to do '+', '-', '*' and '/'

```
Enter number> 9
Value= 9
Enter operator> +
Value= 9 + ?
Enter number> 11
Value= 20
Enter number> 17
Value= 17
Enter number> 23
Value= 23
Enter operator> *
Value= 23 * ?
Enter number> 17
Value= 391
Enter number> 35
Value= 35
Enter operator> /
Value= 35 / ?
Enter number> 8
Value= 4.375
Enter number>
```

## 4 Number Guessing

***Knowledge required**: input and print; if statement and loops. Function for random number. After session 4.*

In this game the computer chooses a secret number for the computer to guess. Here is an example:

```
I have chosen a secret number of 3 digits for you to guess
What's your guess>500
Too high. Try lower.
What's your guess>250
Too low. Try higher.
What's your guess>375
Too low. Try higher.
What's your guess>427
Too low. Try higher.
What's your guess>463
Too low. Try higher.
What's your guess>480
Too low. Try higher.
What's your guess>490
Too high. Try lower.
What's your guess>485
Too low. Try higher.
What's your guess>487
Good guess. Well done!
```

# 5 The Shopping List Program

***Knowledge required**: Loops and arrays. Operations to test whether an item is in a list, and append to or remove from a list. After session 6.*

Here is a transcript from a simple 'shopping list' program. Implement this program:

```
Electronic shopping list
Commands: A to add an item; B to mark an item bought; P to
print
Command>A
New item>drink
Command>A
New item>food
Command>A
New item>books
Command>P
List of purchases: []
List to buy: ['drink', 'food', 'books']
Command>B
Item bought>drink
Command>B
Item bought>DVD
That wasn't on the list. Ignored!
Command>P
List of purchases: ['drink']
List to buy: ['food', 'books']
Command>
```

## 5.1 Discussion and hints.

The program has two lists

- A list of items to buy
- A list of items that have been bought.

The commands are as follows:

| | |
|---|---|
| A | Add an item to the list of shopping. Should you check if the item is already on the list? |
| B | Transfer an item from the shopping list to the list of purchases. What to do if the item wasn't on the list? |
| P | Print out the two lists. |

The program is an endless loop. The loops prompt for a command. The following outline program runs, but does not do much. You may wish to use it as a starting point:

```
shopping = []
purchases = []
while True :
    cmd = input("Command> ")
    if cmd == 'A':
        print("Whoops. No code for adding yet")
    elif cmd == 'B' :
        print("Whoops. No code for buying yet")
    elif cmd == 'P' :
        print("Whoops. No code for printing yet")
    else :
        print("The command are A (add), B (buy), P (print)")
```

## 6 Arithmetic Quiz Using Functions

***Knowledge required**: Loops and functions (though the program could be written without). After session 8.*

The following output shows an example run of the program (user input underlined):

```
What is 0 + 6 ? Your answer: 6
Correct well done!
What is 7 - 1 ? Your answer: 6
Correct well done!
What is 5 * 9 ? Your answer: 45
Correct well done!
What is 2 + 7 ? Your answer: 2
Whoops! That's wrong. Correct answer = 9
```

The program can be implemented using functions. Here is an incomplete outline.

```
import random

# Global variables
op = ""  # operator chosen
op1 = 0  # first operand
op2 = 0  # second operand

#
# Generate a random sum. Assign to the global variables
#
def randomSum(oplist, maxNum):
    global op, op1, op2
    ...

#
# Calculate the correct answer and return it
#
def correctAnswer(operator, left, right) :
    ...
    return ...

#
# Get the user's answer and check it. Return True if it is correct
#
def checkAnswer(correctAns) :
    ...
    return ...
```

Here is the **complete** part of the program that uses the functions.

```
OPERATORS = ["+", "-", "*"] # The possible operators
MAX = 9                     # Max argument

play = True    # Initialise the loop variable
while (play):

    # generate a sum randomly
    randomSum(OPERATORS, MAX)

    # show the sum to the user
    print("What is", op1, op, op2, "?", end=" ")

    # get the correct answer
    correct = correctAnswer(op, op1, op2)

    # check the user's answer
    play = checkAnswer(correct)
```

## 7 The Shopping List Program with Files

***Knowledge required**: As for the original shopping list, but also files. After session 10.*

Some people buy almost the same things every time they go shopping. Enhance the shopping list program so that the shopping list can be saved in a file. It could work like this:

1. When the program starts it offers an options to read 'R' a shopping list from a file. Otherwise an empty list is used.
2. The shopping list can be printed 'P' and items can be added 'A' or deleted 'D'.
3. The amended list can be written 'W' to the file, whose name is entered.
4. The user chooses to start shopping 'S' and can then buy 'B' or print 'P' items as before.

Many other enhancements are possible. For example:

- Also record the number or amount of an item.
- Arrange the list into sections for different shops.

## 8 Mastermind with Numbers

***Knowledge required**: This problem does not require more programming knowledge – loops and arrays are sufficient – but it is harder.*

A simple board uses coloured pegs and requires one player to guess the pattern of pegs chosen by the other. This version uses digits; an example shows how it works:

```
Guess the secret sequence of 4 digits.
Enter guess digits separated by spaces.
 Enter guess>3 4 5 6
  Correct place =  0
  Wrong place =  0
 Enter guess>1 2 7 8
  Correct place =  0
  Wrong place =  2
 Enter guess>1 2 9 9
  Correct place =  0
  Wrong place =  2
 Enter guess>1 2 1 2
  Correct place =  0
  Wrong place =  1
 Enter guess>7 8 9 2
  Correct place =  1
  Wrong place =  2
 Enter guess>7 9 0 2
  Correct place =  2
  Wrong place =  2
 Enter guess>7 0 2 9
  Correct place =  2
  Wrong place =  2
 Enter guess>7 9 2 0
  Correct place =  4
 Well done! You got it
```

Looking to the end, the answer is:
7 9 2 0
It is generated randomly so ever game differs

1 2 9 9 has none correct, but two digits '9' and '2' are in the answer

7 0 2 9 has the '7' and '2' correct but the others are swapped