Version: 01/03/2023 13:11

CSC1015F Assignment 4

Control (if, for)

Assignment Instructions

This assignment involves constructing Python programs that use input and output statements, 'if' and 'if-else' control flow statements, 'for' statements, and statements that perform numerical manipulation.

NOTE Your solutions to this assignment will be evaluated for correctness and the following qualities:

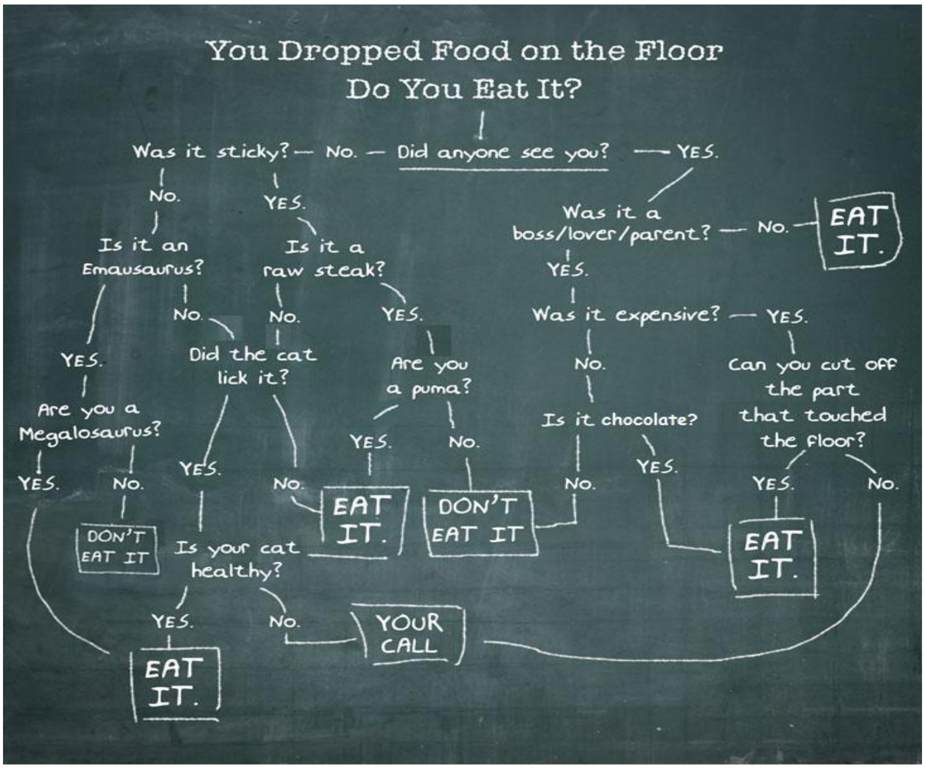
- Documentation
    - Use of comments at the top of your code to identify program purpose, author and date.
    - Use of comments within your code to explain each non-obvious functional unit of code.
- General style/readability
    - The use of meaningful names for variables and functions.
- Algorithmic qualities
    - Efficiency, simplicity

These criteria will be manually assessed by a tutor and commented upon. In this assignment, up to 10 marks will be deducted for deficiencies.

Question 1 [20 marks]

We've all been there: You dropped your cupcake on the ground. Did it land icing up, or down? Can you just scrape off the icing? How many hours have you lost trying to decide?

Using this flowchart, write a program called "cupcake.py" to determine whether or not you should eat the food:



Your program must ask a series of questions to determine if you should eat the food or not. Assume that there are no errors in the input.

This type of program is a simple variant of *artificial intelligence* known as an **expert system** and the flowchart is known as a **decision tree**.

Sample IO *(The input from the user is shown in bold font – do not program this)*:

```
Welcome to the 30 Second Rule Expert
------------------------------------
Answer the following questions by selecting from among the options.
Did anyone see you? (yes/no)
yes
Was it a boss/lover/parent? (yes/no)
yes
Was it expensive? (yes/no)
yes
Can you cut off the part that touched the floor? (yes/no)
no
Decision: Your call.
```

Sample IO *(The input from the user is shown in bold font – do not program this)*:

```
Welcome to the 30 Second Rule Expert
------------------------------------
Answer the following questions by selecting from among the options.
Did anyone see you? (yes/no)
no
Was it sticky? (yes/no)
no
Is it an Emausaurus? (yes/no)
no
Did the cat lick it? (yes/no)
yes
Is your cat healthy? (yes/no)
yes
Decision: Eat it.
```

Question 2 [30 marks]

Write a program called '`perfect.py`' to determine if a given number is a perfect number or not. *A perfect number* is a positive integer that is equal to the sum of its proper divisors. For example, 6 is a perfect number because its proper divisors are `1, 2, 3` and `1+2+3=6`. However, 12 is not a perfect number because its proper divisors are `1, 2, 3, 4, 6` and their sum is not equal to 12.

Your program should print all the proper divisors of a number followed by a statement that prints whether the number is a perfect number or not.

Sample IO *(The input from the user is shown in* ***bold*** *font – do not program this)*:

```
Enter a number:
28
The proper divisors of 28 are:
1 2 4 7 14

28 is a perfect number.
```

Sample IO *(The input from the user is shown in* ***bold*** *font – do not program this)*:

```
Enter a number:
63
The proper divisors of 63 are:
1 3 7 9 21

63 is not a perfect number.
```

Question 3 [25 marks]

Write a program called 'row.py' that asks the user to enter a number, *n*, where *-6<n<93*. The program will print a sequence of 7 numbers, starting from that value.

Each number must be printed using exactly two characters. If the number takes two characters to print, e.g. 34 or -5, then just print it. If the number takes less than two characters to print, e.g. 0 or 9, then print a space in front of it.

Numbers must be separated by a single space.

Sample IO *(The input from the user is shown in bold font – do not program this)*:

```
Enter the start number:
7
 7  8  9 10 11 12 13
```

Sample IO *(The input from the user is shown in bold font – do not program this)*:

```
Enter the start number:
35
35 36 37 38 39 40 41
```

Introducing some terminology, we say that the numbers are printed using a field width of 2 and are right-justified.

Question 4 [25 marks]

Write a program called 'column.py' that asks the user to enter a number, *n*, where *-6<n<2*. Starting from *n*, the program will print out every 7th number in the range *n* to *n+41*.

Each number will appear on a new line.

Numbers are printed using a field width of 2 and are right-justified.

Sample IO *(The input from the user is shown in* ***bold*** *font – do not program this)*:

```
Enter a number:
-5
-5
 2
 9
16
23
30
```

Submission

Create and submit a Zip file called ‘ABCXYZ123.zip’ (where ABCXYZ123 is YOUR student number) containing cupcake.py, perfect.py, row.py and column.py.

NOTES:

1. FOLDERS ARE NOT ALLOWED IN THE ZIP FILE.
2. As you will submit your assignment to the Automarker, the Assignment tab will still say “Not Complete”. THIS IS COMPLETELY NORMAL. IGNORE IT.