CSC1015F Assignment 3

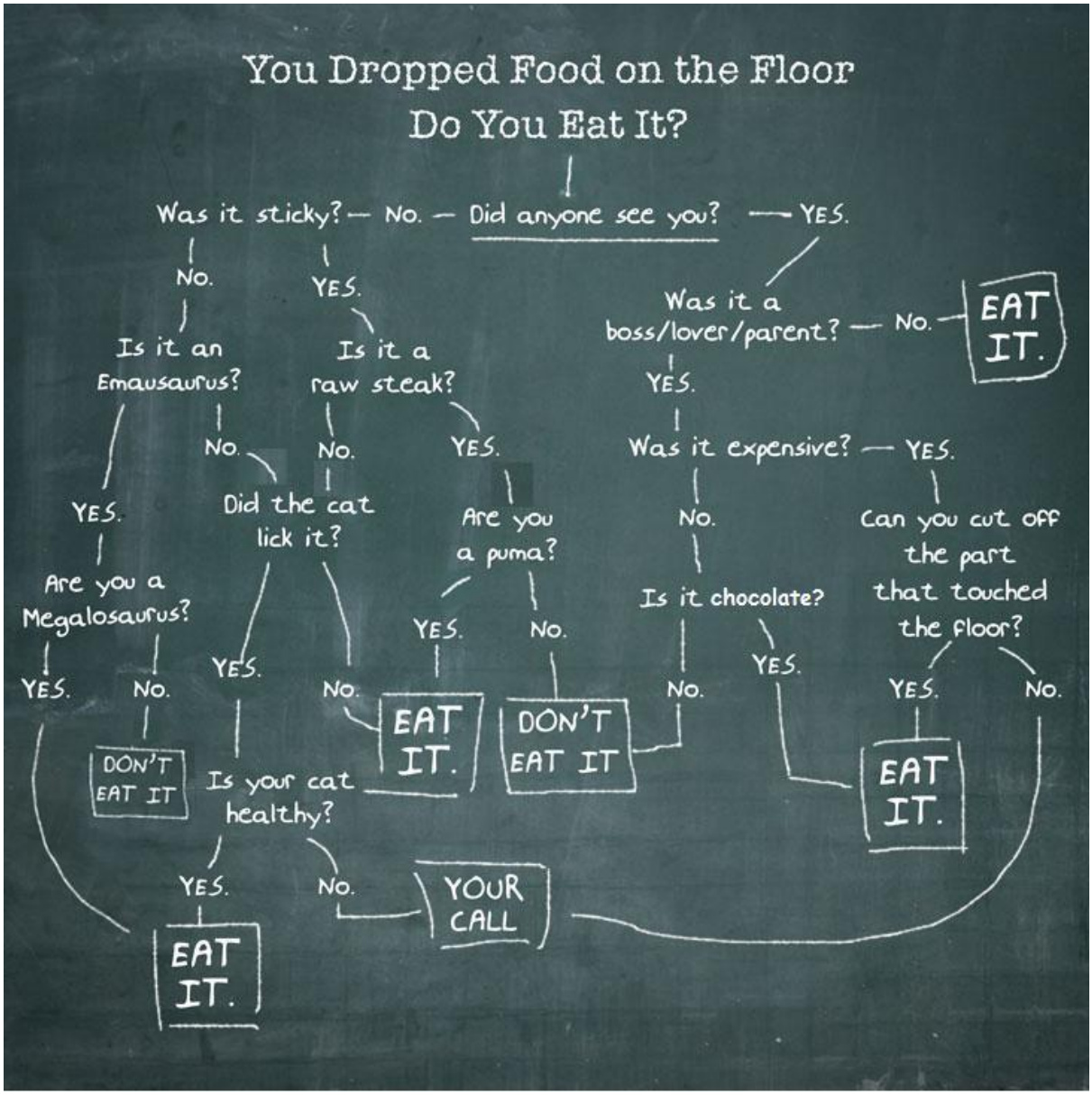
Control (if, while)

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.

Question 1 [30 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? The following flowchart can be used to determine whether or not dropped food can be eaten:



On the Vula page for this assignment you will find a program called '`cupcake.py`'. The program is supposed to implement the flowchart, asking a series of questions to determine if you should eat the food or not.

Sample I/O:

```
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.
```

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.

Unfortunately, though the program consists of suitable statements, they are (i) in the wrong order, and (ii) not correctly indented.

Download the program and organise the statements so that it operates correctly.

Question 2 [20 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:

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

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

Question 3 [20 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 I/O:

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

Question 4 [30 marks]

Write a program called '`grid.py`' that accepts a number, *n*, where *-6<n<2*. The program will print out the numbers *n* to *n+41* as 6 rows of 7 numbers. The first row will contain the values *n* to *n+6*, the second, the values *n+7* to *n+7+6*, and so on.

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

Fields are separated by a single space. There are no spaces after the final field.

Sample I/O:

```
Enter the start number:
-2
-2 -1  0  1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31 32
33 34 35 36 37 38 39
```

HINT: Use a '`for`' loop within a '`for`' loop.

Submission

Create and submit a Zip file called '`ABCXYZ123.zip`' (where ABCXYZ123 is YOUR student number) containing `cupcake.py`, `row.py`, `column.py` and `grid.py`.