“3-a-day” A-Level Exam Practice Unit 2 (002)

Question 1

Convert the following algorithm run-times into Big-O Notation

1. $6n^2 + 8n + 9$
2. 123
3. $9^4 + \log n$

(3 marks)

Question 2

What is the difference between intractable and tractable problems?

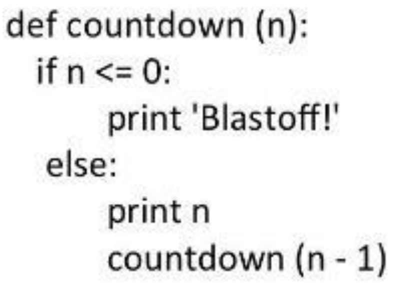
What does the word ‘heuristic’ mean in relation to intractable problems?

(4 marks)

Question3

What is meant by recursion?

Rewrite the following algorithm without the use of recursion.



```
def countdown (n):
   if n <= 0:
          print 'Blastoff!'
    else:
          print n
          countdown (n - 1)
```

(5 marks)

Answer 1

1) O(n²)
2) O(1)
3) O(log n)

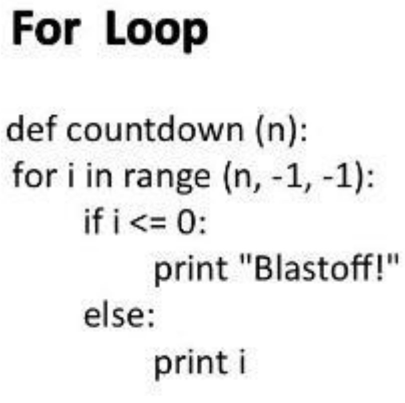
Answer 2

Any problem that can be solved in reasonable time (polynomial or less) is known as a **tractable** problem.
Intractable problems are those which take an extremely long (yet ‘less than infinite’) time to solve

With intractable problems, it is sometimes best to take a **heuristic** approach – where we use our experience to produce a ‘good enough’ solution. For example, finding the shortest distance between an increasing number of locations

Answer 3

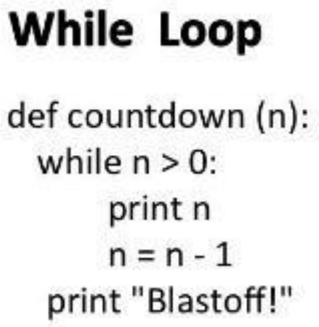
Recursion is when a function will call itself. A recursive function should have a stopping condition, otherwise the function will continue to call itself, adding function call after function call to the stack until the computer can no longer store the function calls – this is known as a stack overflow error.



For Loop

```
def countdown (n):
for i in range (n, -1, -1):
      if i <= 0:
            print "Blastoff!"
      else:
            print i
```

OR



While Loop

```
def countdown (n):
   while n > 0:
         print n
         n = n - 1
    print "Blastoff!"
```