

# EEN218

## Happy Test

### 8th March 2011

Who are you?

What is your student number?

Did you cheat on this test?

Sign that statement.

Don't make any marks in my boxes.

These are my boxes.

Question	1	2	3	4
Out of	33	33	33	1
Grade				

# 1

In simple cases, an array may be created like this:

```
string AAA[1000];
```

but sometimes a declaration in this form is needed:

```
string * BBB;
```

## Short Answers:

- i. What other step is required before `BBB` can actually be used as an array?
- ii. What happens if a program does not take that extra step? Perhaps doing something like this:

```
{ string * BBB;  
  BBB[17] = "easy";
```
- iii. What advantages does the second method (`BBB`) have over the first (`AAA`) - What is it possible to do with `BBB` that can not be done with `AAA`?
- iv. What is a likely cause of a “Segmentation Fault”, and what might a sensible programmer do when faced with one?
- v. What do the C++ operations `new` and `delete` do - What is their purpose?
- vi. What happens if a program uses `new` frequently, but never uses `delete`?
- vii. What is a constructor?
- viii. I want to define a simple struct to represent pet mice. It must have a string to store the mouse’s name, and another string for its colour. It must also have a number to record how much I paid for it, and a true/false value to record whether my cat ate it or not. Additionally, the struct must have a suitable constructor. I am too lazy to do it. Write the complete definition in C++ for me.

## 2

You may have heard of the Collatz Sequence before. It is a very simple thing - take any number, and if it is even divide it by two; if it is odd multiply it by three and add one.

if n is even,  $n = n \div 2$   
if n is odd,  $n = n \times 3 + 1$

Repeat that process over and over again until the number is equal to one.

For example, if you start with 6, the sequence is 6, 3, 10, 5, 16, 8, 4, 2, 1. It is impossible to predict how long the sequence will be. Starting from 31, the sequence is 107 steps long, but starting from 32 it is only 6 steps long.

I have attempted to write a function that works out the sequence from any starting point, puts the entire sequence in an array, and returns that array as a result. Unfortunately, I have got the function wrong:

```
int * collatz(int n)
{ int sequence[];
  int length = 0;
  while (n != 1)
  { sequence[length] = n;
    length = length+1;
    if (n%2 == 0)
      n = n/2;
    else
      n = n*3+1; }
  return sequence; }
```

It is essential that this function should not be *very* wasteful of memory.

Correct the function.

### 3

Reminders: if  $N$  is the amount of data in use,  
a Linear algorithm takes time proportional to  $N$ ,  
a Quadratic algorithm takes time proportional to  $N^2$ ,  
a Cubic algorithm takes time proportional to  $N^3$ ,  
an Exponential algorithm takes time proportional to  $2^N$ ,

a

Here is a very badly designed sorting algorithm (it doesn't even work)

```
for (int i=1; i<N; i+=1)
{ int p = 0, q = N;
  for (int x=1; x<N; x+=1)
  { if (A[x] < A[p])
    p = x; }
  for (int y=1; y<N; y+=1)
  { if (A[y] > A[q])
    q = y; }
  A[p] = A[q];
  A[q] = A[p]; }
```

What kind of algorithm is it (in terms of linear, quadratic, cubic, etc)?  
Explain why.

b

If a Quadratic algorithm takes 10mS (0.01 seconds) to process 1,000 data items, how long should it take to process:

- i. 2,000 items
- ii. 3,000 items
- iii. 1,000,000 items
- iv. 100 items

c

If a Quadratic algorithm takes 1 second to process 1,000,000 data items, and you discover a Linear algorithm that does the same job, roughly how long would your new algorithm take?

d

If a Cubic algorithm takes 10mS (0.01 seconds) to process 100 data items, how long should it take to process:

- i. 1,000 items
- ii. 1,000,000 items

e

If an Exponential algorithm takes 1mS (0.001 seconds) to process 10 data items, how long should it take to process:

- i. 20 items
- ii. 40 items

4

Draw a picture of a cat sitting on a quadratic chair.