COS 212 Tutorial 2: Version B

- 15/02/2016
- 2 questions for a total of 23 marks.

Name and Surname:____

Student/Staff Nr:____

Instructions

1. This tutorial sheet must be completed individually.
2. You are allowed to make use of any material.
3. Answer all of the questions in the space provided.
4. Be sure to hand in your completed sheet at the end of the tutorial class.

Question 1 Complexity .... (9 marks)

1.1 Assume that only assignments to variables are counted. Give the worst case big-O notation complexity for each of the following pseudo code segments.

1.2 (1)

```
for(i = 0; i < n; ++i)
    for(j = 0; j < n; ++j)
        for(m = i-2; m <= i; ++m)
            for(k = 0; k < n; ++k)
                b = i + j + k + m
```

1.3 (1)

```
method(arr[])
    for(i = 1; i < arr.length; ++i)
        el = arr[i]
        for(j = i; j > 0 && el < arr[j-1]; --j)
            arr[j] = arr[j-1]

        arr[j] = el
```

1.4 (1)

```
for(i = 0; i < n; ++i)
    for(j = 0; j < n; ++j)
        for(m = i-2; m <= i; ++m)
            for(k = 0; k < n; ++k)
                b = i + j + k + m
```

1.5 (1)

```
for(i = 0; i < 10; ++i)
    for(j = 0; j < n; ++j)
        k+=6
```

Answer:

1.6 (1)

```
int test(n)
{
     int j = 0;
     for(int i = n; i >= 1; i/=2)
         j*=j+i;
     return j;
}
```

Answer:

1.7 (1)

```
for(i = 0; i < n; ++i)
    for(j = 0; j < n; ++j)
        for(m = i-2; m <= i; ++m)
            for(k = 0; k < n; ++k)
                b = i + j + k + m
```

Answer:

1.8 (1)

```
int i = 0;
int j = n;

while(j != i && i < n)
{
        ++j;
        i=j;
}
```

Answer:

1.9 (1)

```
boolean method(int k, int[] arr, int i, int j)
{
        if(i > j)
                return false;

        if(arr[(i+j)/2] == k)
                return true;

        if(arr[(i+j)/2] < k)
                return method(k,arr,i,((i+j)/2)-1);

        else
                return method(k,arr,((i+j)/2)+1,j);
}
```

Answer:

1.10 (1)

```
int calculation(int n)
{
        if(n <= 0)
                return 0;

        else if(n > 10)
                return n;
```

```
          else
                    return calculation(5 + calculation(5*n));
}
```

Answer:

Question 2 Recursion .... (14 marks)

Consider the following recursive method:

2.1 One can solve the N-queens problem, with N any positive integer, with the following pseudo code: (7)

```
placeQueen(row)
          for (col = 0; col < N; ++col)
                    if board[row][col] is open
                              place queen in board[row][col]
                              if(row < N)
                                        placeQueen(row+1)
                              else
                                        done
                              remove queen from board[row][col]
```

Assuming, starting from the top left, rows and columns are indexed from 0 on a chessboard, use the above algorithm to solve the 6 queens problem. Complete the following table by writing down the column index for each queen:

Queen	**Row Index**	**Column Index**
Queen 1		
Queen 2		
Queen 3		
Queen 4		
Queen 5		
Queen 6		

2.2 Consider the following recursive method: (7)

```
public int method(int x, int y)
{
        if((y <= x) && (x % y == 0))
                return y;

        if(x < y)
                return method(y,x);

        return method(y, x % y);
}
```

Translate this method into an iterative version.