

CS1020 Data Structures and Algorithms I

ANSWER SHEETS

Answers included

INSTRUCTIONS TO CANDIDATES

1. This document consists of **FOUR (4)** printed pages.
2. Write your Student Number and Tutorial group clearly below with a pen.
3. You may use pencil to write your answers.

STUDENT NUMBER: A

--	--	--	--	--	--	--	--	--

TUTORIAL GROUP

--

<i>For examiners' use only</i>		
<i>Question</i>	<i>Max</i>	<i>Marks</i>
Q0	1	
Q1-5	5	
Q6	2	
Q7	3	
Q8	5	
Q9	6	
Q10	9	
<i>Total</i>	30	

MCQs

(Bonus) Q0:

A

Q1.

A

Q2.

D

Q3.

C

Q4.

E

Q5.

B

[5 marks]

Q6.

[2 marks]

3 5

Q7.

[3 marks]

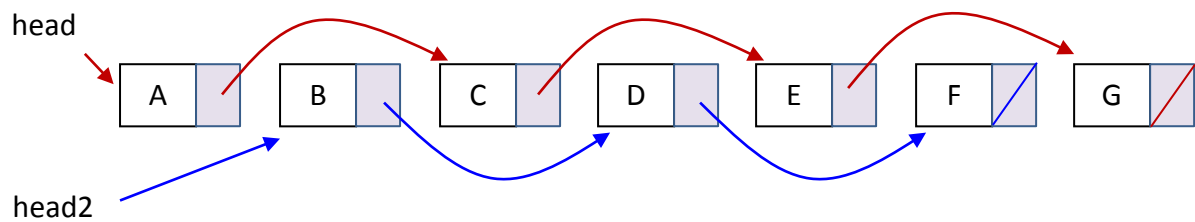
- i. **C.m in A**
- ii. **C.n in C**
- iii. **Error; p() not defined in class B.**

Q8.

[5 marks]

(a) [2 marks]

After executing the algorithm on the given linked list:



(b) [3 marks]

Doesn't work if list contains an even number of nodes:
 'jump' becomes null in the loop → NullPointerException.

Add after line 6:

```

if (jump == null) {
    break; // or curr2 = null;
}
else {

```

Add after line 9:

```

}

```

Q8. (Continue here only if you need more space.)

Q9.

[6 marks]

```
int digit;
for (Integer num: numbers) {
    while (num > 0) {
        digit = num % 10;
        digits.remove((Integer) digit);
        num /= 10;
    }
}
```

Read Aaron's comments document to find out why the cast (Integer) is needed here.

```
int digit, num;
for (int i=0; i<numbers.size(); i++) {
    num = numbers.get(i);
    while (num > 0) {
        digit = num % 10;
        digits.remove((Integer) digit);
        num /= 10;
    }
}
```

Q10.

(a) [4 marks]

```
private double computePerimeter() {
    double sum = 0.0;

    int size = _vertices.size();

    for (int i=0; i<size; i++){
        sum += Math.hypot(
            _vertices.get(i%size).getX() -
            _vertices.get((i+1)%size).getX(),
            _vertices.get(i%size).getY() -
            _vertices.get((i+1)%size).getY());
    }

    return sum;
}
```

(b) [5 marks]

```
public static boolean isValid(ArrayList<Point> pts)
{
    int size = pts.size();
    if (size < 3)
        return false;

    for (int i=0; i<size-1; i++) {
        for (int j=i+1; j<size; j++) {
            if (pts.get(i).equals(pts.get(j)))
                return false;
        }
    }
    return true;
}
```

— END —