INSTRUCTIONS TO CANDIDATES

1. This examination paper contains FIVE (5) questions and comprises SEVENTEEN (17) printed pages, including this page.

2. Answer ALL questions within the space in this booklet.

3. You may use pencil to write your answers.

4. This is an OPEN BOOK examination.

5. Write your Matriculation Number below.

MATRICULATION NUMBER: ____________________________

<table>
<thead>
<tr>
<th>EXAMINER'S USE ONLY</th>
<th>Question</th>
<th>Mark</th>
<th>Score</th>
<th>Graded By</th>
<th>Checked By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q5</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (10 points) What is the output of the following program? (Write your answer on the next page)

```java
class A {
    int f (int n) {
        return n + 2;
    }

    int h (int n) {
        return n + f(n);
    }
}

class B extends A {
    int g (int n, A a) {
        return f( a.f(n) );
    }

    int f (int n) {
        return n * 10;
    }
}

class C extends B {
    int h (int n) {
        return n - 6;
    }
}

public class D {
    public static void main (String[] args) {
        int x = 2;
        int y;
        A a = new A();
        B b = new B();
        C c = new C();
        System.out.println(c.h(2));
        System.out.println(c.f(2));
        x = c.f(x);
        System.out.println(x);
        y = c.g(5, b);
        System.out.println(y);
        System.out.println(b.h(4));
    }
}
```
Answer for Question 1:
2. (15 points) This question is on string processing.
   (a) (5 points) Write a method, \texttt{int myLength (String s)}, that takes a string \texttt{s} as input and returns the length of \texttt{s}. You may use any method in the \texttt{String} class except, of course, the \texttt{length} method.
   (b) (10 points) Write a recursive method, \texttt{String getNthString (String s, int n)}, that takes a string \texttt{s} and a positive integer \texttt{n} as inputs and returns the \texttt{n}th word in \texttt{s}. You may assume the following:
   - there is exactly one blank space between two consecutive words in \texttt{s}; and
   - there are no blank spaces before the first word or after the last word in \texttt{s}.
   For example, if \texttt{s} = “one two three”, then
   - \texttt{getNthString(s,1)} returns “one”.
   - \texttt{getNthString(s,2)} returns “two”.
   - \texttt{getNthString(s,3)} returns “three”.
   - \texttt{getNthString(s,i)} returns an empty string (i.e., “”). if \texttt{i} ≥ 4.

Answer for Question 2(a):
Continuation of answer for Question 2(a):
Continuation of answer for Question 2(b):
3. (15 points) Write a method, \texttt{int}[] \texttt{oddEven} (\texttt{int}[] A), that takes an array of distinct integer values \textit{A} as input, and returns a new array \textit{B} of integers that is equivalent to the input array \textit{A} except that the odd values appeared before the even values in \textit{B}. The ordering among the odd values in \textit{A} must be preserved in \textit{B}; i.e., if an odd value \textit{x} appears before another odd value \textit{y} in \textit{A}, then \textit{x} must also appear before \textit{y} in \textit{B}. Similarly, the ordering among the even values in \textit{A} must also be preserved in \textit{B}.

For example, if the contents of the input array \textit{A} is \{2, 10, 5, 1, 6, 13, 40, 20, 61\}, then the contents of the returned array \textit{B} is \{5, 1, 13, 61, 2, 10, 6, 40, 20\}.

Answer for Question 3:
Continuation of answer for Question 3:
4. (15 points) A two-dimensional matrix $M$ is defined to be *checkered* if (1) $M$ contains at most two distinct values, and (2) the values of any two consecutive entries in $M$ (within the same row or same column) are different; i.e., $M[i][j] \neq M[i][j + 1]$ and $M[i][j] \neq M[i + 1][j]$.

Write a method, `boolean isCheckered (char[][] M)`, that takes a two-dimensional array (of type `char`) $M$ as input, and returns `true` if $M$ is checkered and `false` otherwise.

Answer for Question 4:
Continuation of answer for Question 4:
5. (25 points) The game of chess is played by two players. One player is referred to as the White player and the other player as the Black player depending on the colour of the pieces that they are playing. In a chess tournament, each player plays two games with every opponent, with one game as the White player and the other game as the Black player. For each game played, a player scores 1 point if he wins, 0 point if losses, or 0.5 point if the game is drawn.

Consider a two-dimensional $n \times n$ array of double, \texttt{scores}, which records the game scores for a chess tournament with $n$ chess players (numbered 0 to $n - 1$) such that \texttt{score[i][j]} is the score for the game between player $i$ (as the White player) and player $j$ (as the Black player).

(a) (10 points) Write a method, \texttt{double[][] totalScore (double[][] scores)}, that takes a two-dimensional square matrix of double, \texttt{scores}, as input, and returns a one-dimensional array of double that records the total score for each player. That is, if \texttt{total} is the returned array, then \texttt{total[i]} is the total score of all games played by player $i$.

(b) (15 points) Between two players $X$ and $Y$, we say that player $X$ beats player $Y$ if

1. player $X$ won at least one game against player $Y$; and
2. player $Y$ did not win any game against player $X$.

A player $X$ is considered to be stronger than another player $Y$ if any one of the following conditions hold:

1. The total score for player $X$ is more than the total score for player $Y$.
2. The total scores for players $X$ and $Y$ are the same, and player $X$ beats player $Y$.
3. The total scores for players $X$ and $Y$ are the same, and neither of the players beat the other. The total number of games won by player $X$ is more than the total number of games won by player $Y$.

Write a method, \texttt{int comparePlayer (double [][] scores, int i, int j)}, that takes as inputs, a two-dimensional square matrix of double, \texttt{scores}, and two player numbers, $i$ and $j$; and returns one of the following three values: (1) $i$ if player $i$ is stronger than player $j$, (2) $j$ if player $j$ is stronger than player $i$, or (3) $-1$ if neither of the two players is stronger than the other.
Continuation of answer for Question 5(a):
Answer for Question 5(b):
Continuation of answer for Question 5(b):
Continuation of answer for Question 5(b):

BLANK PAGE

END OF PAPER