**NATIONAL UNIVERSITY OF SINGAPORE**
**SCHOOL OF COMPUTING**

**PROGRAMMING EXAMINATION FOR**
**Semester 1, 2004/2005**

**CS1101: Programming Methodology**

30 October 2004 Time Allowed : 2 hour

---

## INSTRUCTIONS TO CANDIDATES:

(a) Do not read the questions until you are told to do so.

(b) This examination paper contains THREE (3) questions and comprises SEVEN (7) printed pages.

(c) Complete and hand in as many of the following programs as you can. The order of the programs that you hand in is not important.

(d) This is an OPEN-BOOK exam. You may refer to your lecture notes, tutorials, recitation notes and past lab assignments. You are allowed to bring in ONE book (e.g. Cohoon) for the session.

(e) You are allowed to open at most TWO ssh clients to sunfire and ONE Internet Explorer (or other browser) window to the following URL for the Java 1.4.2 API. You are not allowed to refer to any online resources other than the Java API.

`http://java.sun.com/j2se/1.4.2/docs/api/`.

(f) You are not allowed to communicate between each other. If you have any question, you must raise it with the invigilator. All forms of electronic communications (e.g. email, SMS, online chat) are strictly forbidden.

(g) DO NOT ATTEMPT TO CHEAT. All systems will be closely monitored throughout the examination.

(h) Do not disclose or discuss the contents of this PE until after 5pm today.

(i) The PE programs will only be graded for its correctness. If you did not get the full correctness score from the autograder, your lab TA will read and manually allocate marks for program correctness.

(j) Do all your written workings and rough work on the blank pages provided. Your written workings will not graded, but they might help you in the event of plagiarism disputes.

(k) You must return this examination paper, with your matriculation number written on it, at the end of your examination. Otherwise your submissions will not be counted.

(l) Remember to distinguish your essential and non-essential outputs with the '#' prefix character.

**Write your Matriculation number here:** _____

**RETURN THIS QUESTIONS PAPER AT THE END OF YOUR EXAM.**

# 1 Program 1: Plurals (30 marks)

## 1.1 Task

Write a program that reads nouns from the input and outputs their plurals on the basis of these rules:

- If the noun ends in "y", remove the "y" and add "ies".
- If the noun ends in "s", "ch", or "sh", add "es".
- In all other cases, just add "s".

The input from `System.in` consists of multiple lines. Each input line contains exactly one noun. Each noun contains one or more letters. There are no spaces or non-letters in the input. All letters are in lowercase. Print the corresponding plural of each noun to `System.out`.

## 1.2 Sample run

```
stutest:~ % java Plurals
# Enter a noun:
dairy
# The plural form of dairy is:
dairies
# Enter a noun:
boss
# The plural form of boss is:
bosses
# Enter a noun:
dish
# The plural form of dish is:
dishes
# Enter a noun:
bird
# The plural form of bird is:
birds
```

## 1.3 Handing in

Hand in your program electronically through your UNIX account as follows:

```
stutest:~ % ~cs1101ta/bin/cs1handin Plurals <your .java file>
```

# 2 Program 2: Candles (30 marks)

## 2.1 Task

Peter has `n` candles. He burns them one at a time and carefully collects all unburnt residual wax. Out of the residual wax of exactly `k > 1` candles, he can roll out a new candle.

How many candles can Peter burn?

The input contains two integer numbers giving the values of `n` and `k`, each on a line of its own.

Your output should consist of just one integer number giving the maximum number of candles that Peter can burn.

## 2.2 Sample run

```
stutest:~ % java Candles
# Enter n, number of candles:
5
# Enter k, number of residuals to make a new candle:
3
# Peter will burn this number of candles:
7
```

In this example, Peter has an initial set of five candles. He burns three of them to get three residuals and forms a new candle. With the new candle and the remaining two candles in the original set, he burns them to get one more new candle. The total candles he can burn is therefore seven.

## 2.3 Handing in

Hand in your program electronically through your UNIX account as follows:

```
stutest:~ % ~cs1101ta/bin/cs1handin Candles <your .java file>
```

# 3 Program 3: Teams (40 marks)

## 3.1 Task

We need to split a group of `n` (`n` is even) players into two teams. Here's what we do: Line up the players in a straight line. Starting from the first player in the line, count the players while reciting a song that consists of `m` words. The `m`'th player leaves the line and joins "Team A". Repeat the song, now starting with the player who comes after the player who just left. The next player who leaves joins "Team B". Whenever the end of the line is reached, the counting resumes at the beginning of the line. Repeat until all players are assigned to one of the two teams.

The first two lines of the input are the numbers `n` and `m`. The next `n` lines of the input are the names of the players. The input name sequence determines the order in which the players are lined up.

Your output should list all the players in "Team A" followed by "Team B" in the order which they joined the respective teams.

## 3.2 Sample run

Input file, `test01.dat`:

```
6
3
Emily
Hannah
Emma
Ashley
Sarah
Victoria
```

Output:

```
stutest:~ % java Teams < test01.dat
# Team A players:
Emma
Ashley
Sarah
# Team B players:
Victoria
Hannah
Emily
```

In this example, there are six players and the song has three words. The counting starts with Emily — Emily is one, Hannah is two, and Emma is three. Emma leaves the line and joins Team A. The song repeats with Ashley, ends with Victoria, who leaves to join Team B. Going back to the beginning of the line, Ashley leaves next to join Team A, and so on.

## 3.3 Handing in

Hand in your program electronically through your UNIX account as follows:

```
stutest:~ % ~cs1101ta/bin/cs1handin Teams <your .java file>
```

(This page is intentionally left blank)

(This page is intentionally left blank)

(This page is intentionally left blank)

(END OF PAPER)