Java Programming

Final Examination on November 18, 2018
Fall 2018

Department of Computer Science & Information Engineering
National Taiwan University

Problem 1 (30 points) The following program is expected to show $1, 0.9, 0.8, \ldots, 0$ and then "End of program." sequentially. However, this program seems not terminate properly when you execute it. Please modify the program so that it performs as expected. Note that you cannot remove the loop.

```java
public class Problem1 {

    public static void main(String[] args) {

        double x = 1;

        do {
            System.out.println(x);
            x -= 0.1;
        } while (x != 0);

        System.out.println("End of program.");

    }
}
```

Hint: numerical error.

Problem 2 (30 points) Let x be any positive integer. Write a program which sums over all digits of x. For example, $x = 123456$. Then the output is 21. You cannot enter x by entering single digits sequentially (using nextInt() by couples of times). *Hint: division, remainder, loop.*

Problem 3 (30 points) Let a and b be two positive integers. Write a program which calculates the greatest common divisor (gcd) between a and b by recursion. You may consider the Euclid's algorithm. *Hint: gcd.*

Problem 4 (40 points) Write two functions for the following requirements:

- Convert a positive decimal number to a binary number.
- Convert a binary number to a positive decimal number.

For example, the decimal integer 13 is equal to the binary number “1101” (represented in a string or a number). *Hint: division, remainder, loop.*

Problem 5 (40 points) Write a program which outputs the following pattern:

```
Enter n? 5
1
1,1
1,2,1
1,3,3,1
1,4,6,4,1
1,5,10,10,5,1
```

Hint: Pascal triangle.

Problem 6 (40 points) Write two programs which check if the string reads the same backward as forward. One is using a loop, and the other using recursion. For example, it will output true if the input string is like “radar”, “noon”, and “anna”. You may use the method **charAt**() of String objects. Note that you cannot solve this problem by reversing the input string. *Hint: recursion, loops.*

Problem 7 (50 points) The number e is a mathematical constant, approximately equal to 2.71828. Write a program to estimate e by using Monte Carlo simulation. It can be done as follows. For each iteration, find the minimal number n so that $\sum_{i=1}^{n} r_i > 1$ where r_i is the random variable following the standard uniform distribution (you can simply use **Math**.random()). Then e is the average of n. *Hint: Monte Carlo.*

Problem 8 (50 points) Write a program as follows:

- First generate an integer array (say 10000 elements, each number ranging from -2^{31} to $2^{31}-1$; you may use **Integer**.MIN_VALUE and **Integer**.MAX_VALUE, respectively).

- Then output the greatest product of 3 integers among these integers. (Note that it is not necessary to be the product of 3 consecutive integers in the set.)

Make your program run faster than the naive approach which runs in $O(n^3)$ time.

Problem 9 (50 points) Given a string containing just the characters ’(’, ’)’, ’{’, ’}’, ’[’ and ’]’, determine if the input string is valid. An input string is valid if:

- Open brackets must be closed by the same type of brackets.

- Open brackets must be closed in the correct order.

Note that an empty string is also considered valid. Download the sample program from `https://www.csie.ntu.edu.tw/~d00922011/java/299/code/Problem9.java`. For example,

```
: true
() : true
()()() : true
{[(){}][((()))]} : true
( : false
] : false
([)] : false
```

Hint: stack.

SUBMISSION

1. Send your source codes to d00922011@csie.ntu.edu.tw with the mail title: [JavaNNN] your full name, where NNN is the class number.

2. Do not attach the executable files.

3. If you have any suggestion for improving this class, feel free to leave your feedback in the e-mail.

Congratulations!!!

"To start, you don't have to be good;
to be good, you have to start."
– Slogan of NTU Toastmasters