Midterm Practice Problems

Part 1. Show your work for partial credit

(a) Consider the following code segment:

```java
int p = 30;
int b;
Scanner kbrd = new Scanner(System.in);
System.out.print("Please enter an integer");
b = kbrd.nextInt();
if (b > p || b < 0) {
p = 2*p;
b = b/25;
} else {
p = 0;
b = b%2;
}
```

Assume the user entered 55.
What is the value of variable \(p\) after the execution of this program segment?

What is the value of variable \(b\) after the execution of this program segment?

(b) What will be printed by the following code segment:

```java
int a = 35;
int b = 100;
boolean flag = false;
if (!flag || a > b) {
    System.out.println("One");
} else {
    System.out.println("Two");
}
```

(c) Consider the following code segment and show what is printed when it is executed. Show the intermediate values of variables \(k\) and \(s\) for partial credit.

```java
int k = 1;
int s = 0;
do{
    System.out.print(‘*’);
    if (k % 2 == 1)
        s = s + k;
    k++;
}while (s < 5);
System.out.println(k);
System.out.println(s);
```
(d) Consider the following code segment:

```java
Scanner kbrd = new Scanner(System.in);
String myStr = kbrd.nextLine();
int len = myStr.length();
String result;

if (len <=3 || myStr.charAt(0) == 'Z') {
    result = "RARE";
} else {
    result = "NOT RARE";
}
```

Show values of variables `len` and `result` after the execution of this program segment for the following user inputs:

- user enters Zanzibar
  - `len`: ____________  `result`: ____________
- user enters Moo
  - `len`: ____________  `result`: ____________

(b) What will be printed by the following code segment?

```java
int a = 20;
int b = 100;
boolean flag = false;

if ( b % a == 0 )
    flag = true;
if ( !flag ) {
    System.out.println("One");
}
else {
    System.out.println("Two");
}
```

Show values of following expressions:

- `b % a`: ________________________________
- `!flag`: ________________________________

2. In this problem you will develop a main method that prints out a character corresponding to a grade value that is entered by the user. The following chart describes the correspondence between a numeric grade and its letter equivalent.

<table>
<thead>
<tr>
<th>Letter grade</th>
<th>Grade Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90 &lt;= grade</td>
</tr>
</tbody>
</table>

```java
90 <= grade  Letter grade
A
```
3. In this problem you will develop a main method that prints value true if in a word that is entered by user, the first character is the same as the last character, and false otherwise. Your program should be case-sensitive, i.e. consider the capital letters different from lowercase.

For example, when passed the word “Halloween”, the program must print false because the first character is ‘H’ and the last is ‘n’, but when passed the word “classic”, it must print true, since both the first and the last characters are ‘c’.

4. Write a complete program that lets a user enter words, computes how many words in the input have the same starting and ending letter.

The program must at first read the number of words in the list (n) followed by exactly n words each appearing on a separate line. Then, the program must compute and print how many words start with the same letter as they end with.

For example, given the following user input:
5
Shine
on
you
crazy
diamond

(where 5 is the number of words that follow) the program must print

**Number of words with matching first and last letter is 1**

since there is only one word (diamond) with first letter matching the last, and the word “on”, for example, is shorter than the previous word “Shine”.

Here’s another example. For user input
3
one
seven
eighteen

the output should read

**Number of words with matching first and last letter is 0**

Your program doesn’t have to contain comments, but must otherwise use good programming style.

6. Write a complete program that lets the user balance a checkbook.

The program must read the value of the original balance, followed by a sequence of transaction type code (‘w’ or ‘W’ for a withdrawal, ‘d’ or ‘D’ for a deposit) and transaction amount pairs. The program must terminate when either the balance becomes negative or the user enters character ‘S’ (for Stop) instead of a valid transaction code.

The program should output

1. the final balance,
2. the total amount of money deposited,
3. the total amount of money withdrawn.

The following sample interactions demonstrate the desired behavior of the program: User input is indicated in boldface.

Please enter the original balance: 300.25
Please enter transaction type: W
Please enter amount: 20.25
Your updated balance is $280.0

Please enter transaction type: d
Please enter amount: 50.35
Your updated balance is $330.35

Please enter transaction type: W
Please enter amount: 10
Your updated balance is $320.35.

Please enter transaction type: S

PRINTING FINAL VALUES:
BALANCE: 320.35
TOTAL DEPOSITED 50.35
TOTAL WITHDRAWN 30.25

Here’s another example in which the program terminated when the balance dropped below zero.

Please enter the original balance: 300.25
Please enter transaction type: D
Please enter amount: 200.25
Your updated balance is $500.50

Please enter transaction type: d
Please enter amount: 50.35
Your updated balance is $550.85

Please enter transaction type: W
Please enter amount: 600
Your updated balance is $-49.15

PRINTING FINAL VALUES:
BALANCE: -49.15
TOTAL DEPOSITED 250.60
TOTAL WITHDRAWN 600.0