TOBB ETÜ, BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL 113/012 Computer Programming I

HOMEWORK 4 (30 Points)

Nov 15, 2020

1 [10 POINTS] ADDING AND SUBTRACTING LARGE INTEGERS

In this task, you are going to write a Java program which can add and subtract large integers (numbers that cannot be represented using long). For subtraction, you can assume that the first number is larger than or equals to the second number.

EXAMPLE 1
Please enter two numbers. First number:100000000000000000000000
Second number:999999999999999999999999
Please enter 1 for addition or 2 for subtraction:2
Result:1

EXAMPLE 2
Please enter two numbers. First number:98765432198765432198765432198765432 1
Second number:123456789123456789123456789123456789
Please enter 1 for addition or 2 for subtraction:2
Result:864197532864197532864197532864197532

EXAMPLE 3
Please enter two numbers. First number:98765432198765432198765432198765432 1
Second number:123456789123456789123456789123456789
Please enter 1 for addition or 2 for subtraction::1
Result:11111111111111111111111111111111111110

EXAMPLE 4
Please enter two numbers. First number:5456231546513212645121456465 1
Second number:456451214564515645114451
Please enter 1 for addition or 2 for subtraction:1
Result:5456277191634669096685967 9102

EXAMPLE 5
Please enter two numbers. First number:5456231546513212645121456465 1
Second number:456451214564515645114451
Please enter 1 for addition or 2 for subtraction:2
Result:5456185901391756193556945 0200

2 [10 POINTS] DEPT PAYMENT

You have just purchased a new laptop for your Bil113 course. Laptop costs $1,500 on the following credit plan: no down payment, an interest rate of 18% per year (and hence 1.5% per month), and monthly payments of $50. The monthly payment of $50 is used to pay the interest, and whatever is left is used to pay part of the remaining debt. Hence, the first month you pay 1.5% of $1,500 in interest. That is $22.5 in interest. So, the remaining $27.50 is deducted from your debt, which leaves you with a debt of $1472.50. The next month, you pay interest of 1.5% of $1472.50, which is $22,0875. Hence, you can deduct $27.9125 (which is $50–$22,0875) from the amount you owe. This goes on until you have no debt.
In this task you will write a program that tells you how many months it will take you to pay off a loan with the given interest under equal payments each month. Also, you will calculate the total payment paid. Use a loop to calculate the amount of interest and the size of the debt after each month. (Your final program need not output the monthly amount of interest paid and remaining debt, but you may want to write a preliminary version of the program that does output these values.) Use a variable to count the number of loop iterations and hence the number of months until the debt is zero. You may want to use other variables as well. The last payment may be less than the given monthly payment if the debt is small, but do not forget the interest! If you owe $50, your monthly payment of $50 will not pay off your debt, although it will come close. One month's interest on $50 is only 75 cents.

You may assume that monthly payments are greater than the first month's interest. You should use current format as in the last homework while printing the total amount paid.

EXAMPLE 1
Please enter the initial debt:1000
Please enter the yearly interest rate (in percentages):24
Please enter the monthly payment you want to make:50
It takes 26 months to pay the debt and the total payment equals to $1,289.87

EXAMPLE 2
Please enter the initial debt:1000
Please enter the yearly interest rate (in percentages):12
Please enter the monthly payment you want to make:50
It takes 23 months to pay the debt and the total payment equals to $1,121.35

EXAMPLE 3
Please enter the initial debt:1000
Please enter the yearly interest rate (in percentages):12
Please enter the monthly payment you want to make:25
It takes 52 months to pay the debt and the total payment equals to $1,283.47

EXAMPLE 4
Please enter the initial debt:1000
Please enter the yearly interest rate (in percentages):12
Please enter the monthly payment you want to make:12.5
It takes 162 months to pay the debt and the total payment equals to $2,021.85

EXAMPLE 5
Please enter the initial debt:1600
Please enter the yearly interest rate (in percentages):18
Please enter the monthly payment you want to make:25
It takes 217 months to pay the debt and the total payment equals to $5,404.96

3 [10 POINTS] SIMPLIFIED TOWER OF HANOI

In this task, you will write a Java program that allows users to play a simplified version of the classical game called the Tower of Hanoi. In this game, there are three pegs named A,B, and C and three disks with different sizes. At the start of the game, all three disks are stacked on peg A such that the smallest disk is at the top and the largest disk is at the bottom as depicted in Figure 3.1. In order to complete the game, you must move all disk to peg C. The final state of the disks is depicted in Figure 3.2. The rules are as in the following:

- You can only move one disk at a time,
- A disk cannot be placed on a smaller disk.

You will represent the disks with the numbers 1,2, and 3. Smallest disk is represented with 1 and the largest disk is represented with 3. For the visual representation of a state of the game, you will print the disks in each peg on top of each other. You should use "$\backslash t \backslash t$" to arrange the spacing between the pegs. Print one empty line before and after printing the configuration of the game. Notice that printing the state of the game consist of 5 lines. Ask the user to make a move after printing the current state of the game. User must input two pegs separated by "-", e.g., "A-B", "B-C" etc. If the move is legal, move the disk to the desired peg and print out the new state of the game and ask for a new move. If the move is not legal (illegal), print why the

Figure 3.1: Start of the game

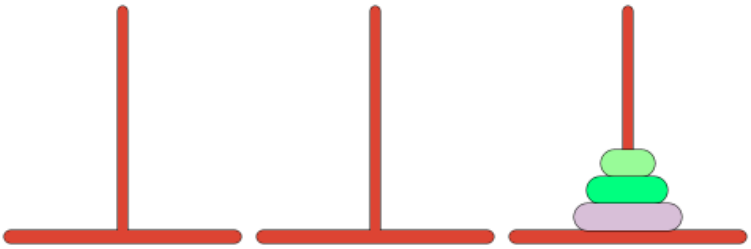

Figure 3.2: End of the game

move is illegal. It can be one of the following: the peg before the dash is empty or user tried to move a disk onto a smaller disk. Print out current state of the game and ask for a new move. Game only ends after all disk moved to the peg C as in the figure 3.2. After the game ends, print the number of legal moves the user made and the number of illegal moves the user tried. You can examine the example below for more clarity. There are two more examples under the resource section in text files.

Game Example:

```
Welcome to the Game of Hanoi!..

1
2
3

Please make a move:A–C


2
3                                   1

Please make a move:A–C
Illegal move: bigger disk cannot be placed on top of smaller disk


2
3                                   1

Please make a move:A–B



3                   2               1

Please make a move:A–B
Illegal move: bigger disk cannot be placed on top of smaller disk



3                   2               1

Please make a move:A–C
Illegal move: bigger disk cannot be placed on top of smaller disk



3                   2               1
```

```
Please make a move:C–B


                1
3               2

Please make a move:A–C


                1
                2               3

Please make a move:B–A



1               2               3

Please make a move:B–C


                                2
1                               3

Please make a move:A–C

                                1
                                2
                                3

Congratulations you won!..
You made 7 legal moves to win.
Also you tried to make 3 illegal moves.
```