Assignment #1 – Due December 10th

**Algorithms**

**Description:** The assignment can be done **individually** or in teams of **two**. Submit one assignment per team of two via Omnivox and NOT MIO. Assignments sent via MIO will be deducted marks. Assignments must be done alone or in groups and collaboration between individuals or groups is strictly forbidden. Submissions that are late will receive a 10% penalty per day, and after four days of no submission a mark of zero will be given.

**General Guidelines When Writing Your Programs:**

Include the following in comments at the top of your program:

// -----------------------------------------------------

// Assignment (include number)

// Written by: (include your name(s) and student ID(s))

// Short Description of your project/code and how you designed it.

// -----------------------------------------------------

Throughout your program, include comments in your program describing the main steps where necessary.

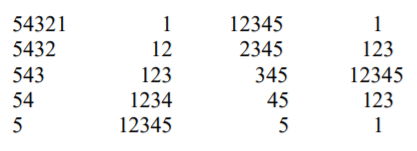
**Printing Patterns (25 points)**

**Description:** Write a Java program that prints one of the following patterns based on the user choice of a pattern number, which must be between 1 and 4, or 5 to quit (See examples below), and an input value, which must be larger than 0 and smaller than 10, and according to the following:

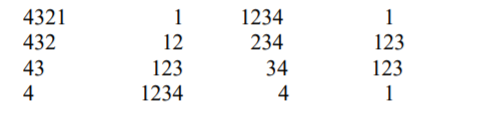
A) If the user enters any invalid pattern number, then the program should display a message indicating that input was invalid and request the user to either enters a correct pattern number or 5 to quit the entire program. That is, an entry of 5 would simply terminate the program.

B) Once the pattern number is correctly supplied, the program requests the user to enter the input value. If the user enters any invalid value that is not within the expected range, then the program should reject this entry and asks the user to re-enter another value; this would repeat indefinitely until a good value is entered.

C) Upon the entry of a good input value, the program must check whether this value is odd or even. If the user enters an odd number, then the program would draw a pattern that is similar to the following (i.e. the shown pattern is drawn if the user enters 5. You should notice that this is only an example; your program must allow for the general case for different appropriate values as indicated above.) You should notice that the patterns are actually quite similar whether the input is odd or even, with the exception of pattern # 4. See below:



If the user enters an even number, the pattern would look as one of the following (i.e. the shown pattern is drawn if the user enters 4. Again this is just an example; your program must work for the general case.)



**Restrictions:**

• You can use any combinations of the selection and repetition statements that we have seen in class. You must however write the needed code yourself (i.e. do not use any built-in methods that may perform what is needed.)

**General Algorithm:**

In general your program should:

1. Display a welcome message with your name(s) on it (i.e. “Welcome to Mike and Linda Triangle/Diamond Printer”).

2. Prompt the user for the pattern they want to display. Make sure the pattern requested is a legal one.

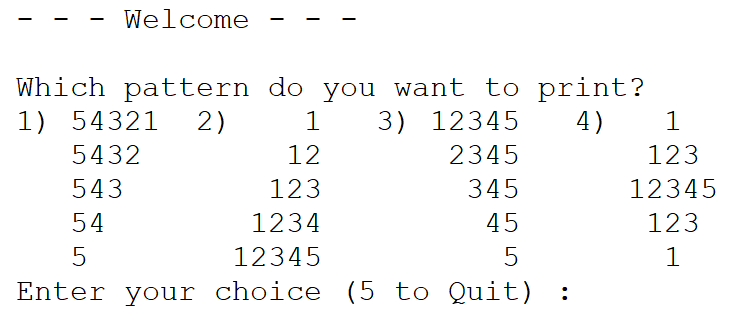
3. Prompt the user for the input value. The input value should be greater than 0 and less than 10. Keep prompting the user until the input value is valid.

4. Display the requested pattern.

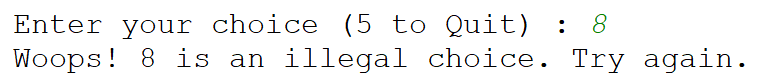
5. As long as the user wants to display another pattern your program repeats steps 2 to 4

6. Display a closing message

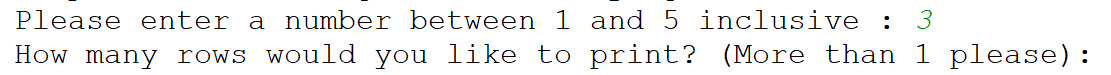
Display a welcome message and the options to choose



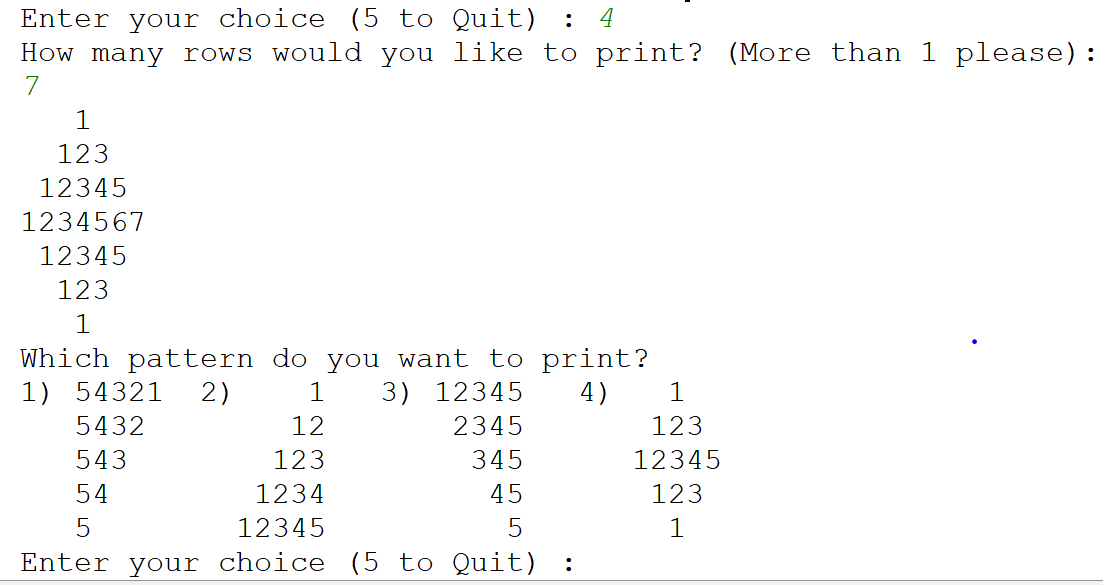
Incorrect choices will result in a message allowing the user to choose again:



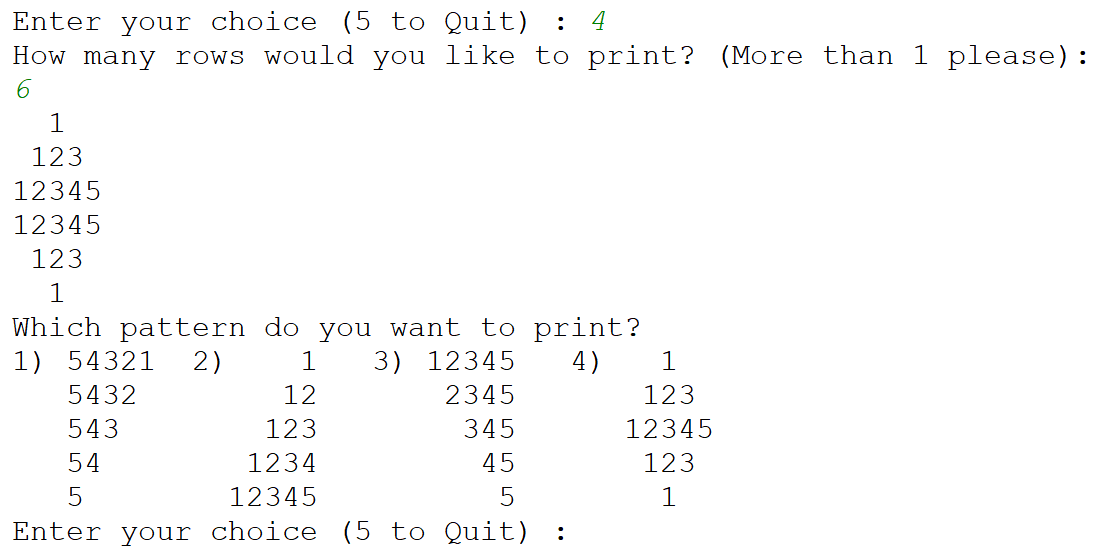
After correct number is chosen ask the user for the number of rows to print:



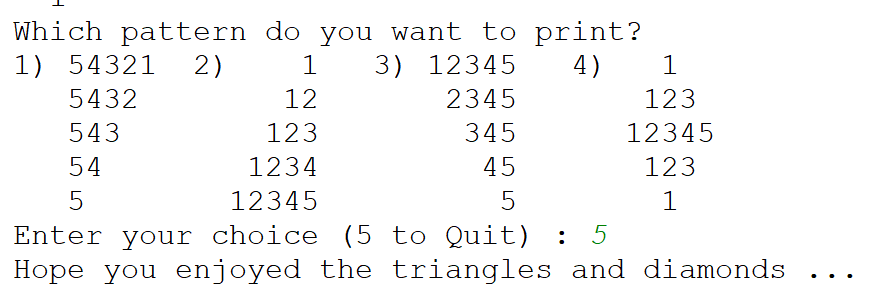
Users should be able to return to the original menu after the pattern is printed:



Pattern (4) should display the triangle differently depending if the number of rows entered is an even number or an odd one:



Include a leaving message for when the user wants to quit the application:



Grading Grid:

|  |  |  |
| --- | --- | --- |
| **Common Criteria for Questions 1,2,3 – (2 pts.)** | | |
| Comments & Programming Style which includes:  - Description of the program (authors, date, purpose)  - Significant names for variables  - Description of variables  - Description of the algorithm | 2 | pts. |
| **Question 1 – (23 pts).** | | |
| No bugs occur | 5 | pts. |
| Menu operation/error checking | 1 | pt. |
| Clarity of output | 2 | pts. |
| Pattern 1 | 2 | pts. |
| Pattern 2 | 3 | pts. |
| Pattern 3 | 3 | pts |
| Pattern 4 | 6 | pts |
| Program loops | 1 | pts |
| **Total 25 pts** | | |