CSC 340 HOMEWORK 2

PROFESSOR GODFREY MUGANDA

1. The Homework Problems

Open up the Netbeans shell provided for you and complete the skeleton program in there to solve the following problems.

1. (Combinations, matrix edition) Write a program that asks the user to enter two integer n and k where $0 \leq k \leq n$. The program builds an array with $n + 1$ rows (you can make it a square matrix if that is easier for you) and fills it in with the part of Pascal's triangle, but only that part that is needed to compute $\binom{n}{k}$.

Print the array. Here is a sample run.

```
This will compute the combinations of n things
taken k at a time where 0 <= k <= n.
Please enter n followed by k :
6 4
The table of comb(n, k) is:

   1
   1   1
   1   2   1
   1   3   3   1
   1   4   6   4   1
   1   5  10  10   5
   1   6  15  20  15
```

You can use the statement

```
      System.out.printf("%4d", comb[r][c]);
```

to print an array entry right-justified in a column that is 4 spaces wide

2. (Combinations, list edition) Do the same problem as in the previous problem, except this time to not use an array. Instead, use to lists. Keep a row of the Pascal's triangle in one list, and use it to build a new list that represents the next row. After that, drop the previous row and use the new row as to create another row, and so on. Unlike arrays, you can print an entire list. Print each new row as it is created.

Make the rows only as long as needed. Here is a sample run

```
This will compute the combinations of n things
taken k at a time where 0 <= k <= n.
Please enter n followed by k : 6 4
The table of comb(n, k) is:

[1]
```

```
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5]
[1, 6, 15, 20, 15]
```

3. Consider this problem:

Determine the number of ways to choose 5 numbers from the first 18 positive integers so that any two chosen numbers differ by at least 2.

The answer is 2002.

More generally, Let $P(n, k)$ be the number of ways to choose k numbers from the first n positive integers so that any two chosen numbers differ by at least 2.

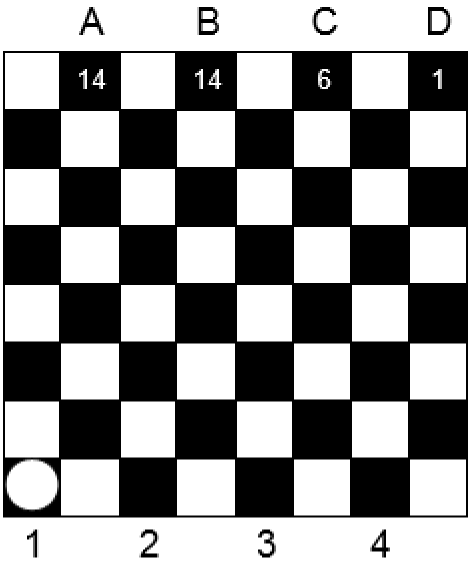
Develop a recurrence relation and boundary conditions for $P(n, k)$ using the same reasoning used to develop the recurrence relation for $\binom{n}{k}$.

Add a file, Problem 3.java, to the project file and use it to solve this problem. The the inputs to the problem and the output should be similar to the first two problems.

4. (Due to Don Nohl, Benedictine University) Paths on a checkerboard:

Add another file to the project, Problem4.java, and use it for the solution of the following problem:

If a checker is placed on one of the four black squares in the first row of an otherwise empty checkerboard, it can move (by a standard checker moves) to any of the four black squares on the last (eighth) row by a variety of different paths. The following checkerboard illustrates the number of different paths that a checker starting in the lower right hand corner (labeled as 1) could take to end up on each of the black squares in the eighth row (labeled as A, B, C, and D). A standard checker move is one square forward on the diagonal. Therefore, a checker may never move onto a white square.



One pair of starting and ending squares is joined by a maximum number of different routes. The purpose of this problem is to identify these two squares by number and letter and give the number of different ways the checker can move from one to the other. Here is the clinker. Your program must be able to solve this problem for a checkerboard of any size from 4 to 32; that is, from 4x4 boards up to 32x32 boards. The lower left-hand corner of the checkerboard is always black.

INPUT: Your program should accept an integer representing the size of the checkerboard at the prompt: "Enter size: ". Your program should continue to accept integers until a zero (0) is entered. At this point your program should terminate without further output.

OUTPUT: Your output should be of the form: number-letter : n different paths where *number* is the number of the starting black square given that the black squares in the first row are numbered starting with 1, and *letter* is the letter of the ending black square given that the black squares are lettered in the last row starting with A, and n is the number of different paths.

Here is a sample interaction:

```
Enter size: 4
2-A: 3 different paths
Enter size: 8
3-B: 35 different paths
Enter size: 32
9-H: 300540195 different paths
Enter size: 0
```

2. Reminder of Ethics Policy

You can get help from each other and from the instructor on these problems, but all help must be at the level exchanging ideas and approaches. Every one should work independent of everyone else on implementation and no one should look at somebody else's code.

3. Due Date

This is due by Saturday night at end of Week 4. However, you should do these problems, or know how to do them, by thursday when the midterm is administered.

For submission, submit the entire project folder.