Assignment 3

Case Studies

1. Formulating the Problem

1.1 Problem Description

Design and implement a Java program that uses recursive pattern to compute how much weight is on each person's back.

1.2 Verbalization

What is the goal?

Create a program to print out human pyramid using recursive function.

What are the givens?

Human weight

Total numbers of rows

1.3 Information Elicitation

Goal

To compute total weight of pyramid

Use recursive function

Givens

Human weight
Total numbers of rows

Unknowns

None

Conditions

Recursive call

2. Planning the Solution

2.1 Solution Strategy

Create a recursive function
Collect user's input
Calculate total weight
Output result on the screen

2.2 Goal Decomposition

Sub-goal 1

Get data from the user.

Sub-goal 2

Collect user's input

Calculate total weight

Sub-goal 3

Display data.

2.3 Resources

Relevant formulas

2.4 Data Organization and Description

Input (givens):

Name	Description	Origin	Used in Sub-goal #
Weight	Weight of person	User	1
Number of Rows	Number of Rows	User	1

Output (unknowns):

Name	Description	Origin	Used in Sub-goal #
Weight	Weight of person	User	1
Number of Rows	Number of Rows	User	1

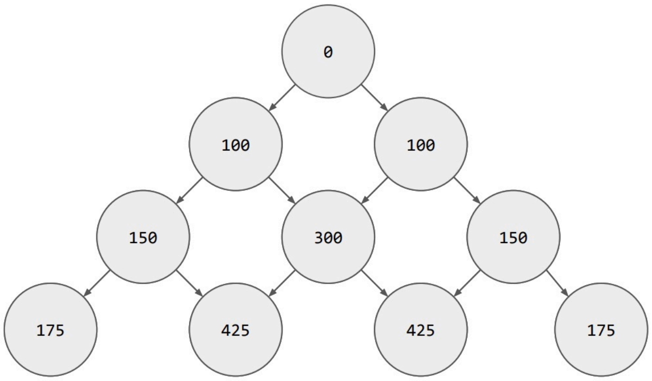
3. Designing the Solution

3.1 Structure Chart

First Level Decomposition

The first level decomposition includes three main goals of this program.

1. Get user’s input
2. Calculate the total weight
3. Display data.



Goal Refinement

Sub-goal 1

Get data from the user.

Sub-goal 1.1

Create class that includes all required fields.

Sub-goal 2

Calculate total weight

Sub-goal 2.1

Implement Interface

Sub-goal 3

Display results.

Sub-goal 3.1

Create Main method to display to display order.

3.2 Module and Data Specifications

Name: Prompt user to enter a full name.
Input: Number of rows
Output: None
Logic: Store user's input
Name: Display for user to choose data

3.3 Algorithm

Logic

1.0 Ask user for number of rows
2.0 Get values from user's input.
3.0 Capture user inputs in variables.
4.0 Calculate the total weight
5.0 Display results

Algorithm Description

Create and display a human pyramid.
Capture users input and store them in variables. Calculate the order total weight of human pyramid and display it on the screen.

4. Translation

4.1 Source Code

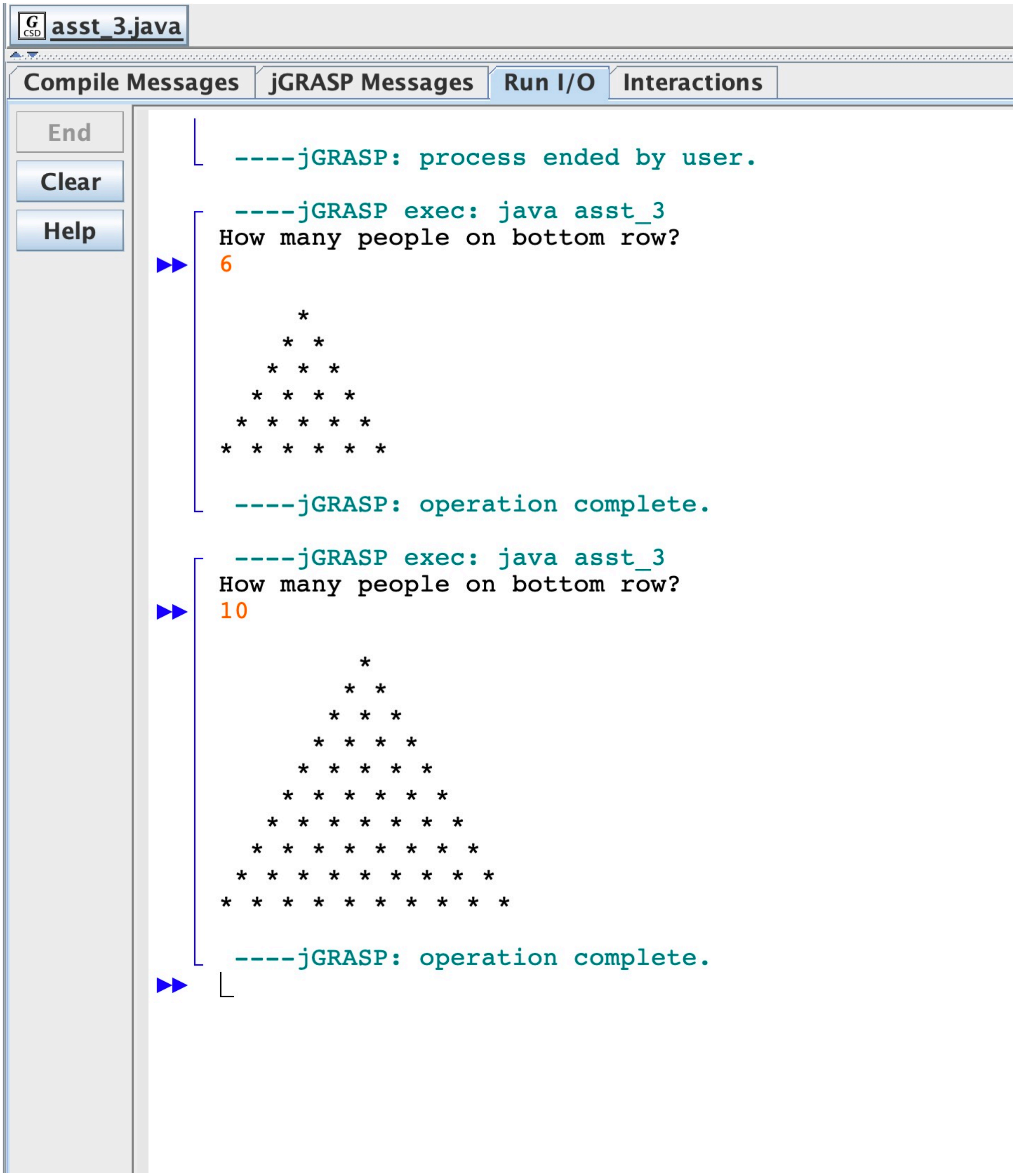
```
//=====================================================
// Name       :  Tsagan Garyaeva
// SID        :  31483539
// Course     :  IT-114
// Section    :  452
// Instructor :  Maura Deek
```

```java
// T.A        :
//=====================================================
//=====================================================
// Assignment # :  3
// Date         :  03/22/2019
//=====================================================
//=====================================================



import java.util.Scanner;

public class asst_3 {

    static void weightOnBackOf(int row, int col)
    {
        if (row < 0) // Base condition
            return;

        // Recursive call
        weightOnBackOf(row - 1, col + 1);

        int i;
        for (i = 0; i < col; i++) // it makes spaces
            System.out.printf(" ");
        for (i = 0; i < row; i++) // for print *
            System.out.print("* ");

        System.out.printf("\n"); // for next line
    }





    public static void main(String[] args) {
    Scanner input =  new Scanner(System.in);
    System.out.println("How many people on bottom row? ");
    int row = input.nextInt();
    weightOnBackOf(row, 0);
    }

}
```

4. Solution Testing

Test the program with following data domain:
The domain range includes integers

Test the program with following data:



```
 ----jGRASP: process ended by user.

 ----jGRASP exec: java asst_3
How many people on bottom row?
6

     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *

 ----jGRASP: operation complete.

 ----jGRASP exec: java asst_3
How many people on bottom row?
10

         *
        * *
       * * *
      * * * *
     * * * * *
    * * * * * *
   * * * * * * *
  * * * * * * * *
 * * * * * * * * *
* * * * * * * * * *

 ----jGRASP: operation complete.
```