Problem Set - 2

Topic: Loops

Section 1: for, while, and 1D Lists

<table>
<tr><th>SL</th><th>Problem</th><th>Difficulty</th></tr>
<tr><td>1</td><td>Write a program to take a number as user input. Next, print all the numbers in ascending order up to that number (print the series up to Nth terms) in a single line.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>2</td><td>1 2</td></tr>
<tr><td>5</td><td>1 2 3 4 5</td></tr>
<tr><td>11</td><td>1 2 3 4 5 6 7 8 9 10 11</td></tr>
</table>
</td><td>*</td></tr>
<tr><td>2</td><td>Write a program to take a number as user input. Next, print all the numbers in <b>descending order</b> up to that number (print the series up to Nth terms) in a single line.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>2</td><td>2 1</td></tr>
<tr><td>11</td><td>11 10 9 8 7 6 5 4 3 2 1</td></tr>
</table>
</td><td>*</td></tr>
<tr><td>3</td><td>Write a program to take a number <b>N</b> as user input and generate a binary number in one line. The length of the number is <b>N</b>. (print 0s and 1s up to <b>N</b>).
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>2</td><td>10</td></tr>
<tr><td>5</td><td>10101</td></tr>
<tr><td>11</td><td>10101010101</td></tr>
</table>
[<b>HINT:</b> Find the pattern. Look at the indexes <b>carefully.</b> What pattern do they follow for each test case]
</td><td>*</td></tr>
</table>

<table>
<tr>
<td>4</td>
<td>
Write a program that will take the number <b>N</b> as user input. Next, the user will enter those <b>N</b> numbers and compute their average.

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>3
10 20 30.5</td><td>Average: 20.1666….</td></tr>
<tr><td>2
22.4 11.1</td><td>Average: 16.7500….</td></tr>
</table>
</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>
Write a program that takes two numbers <b>x</b> and <b>y</b> as user inputs. The inputs must be taken in such a way that the value of y must be greater than the value of x (<b>y &gt; x</b>). Next, the program will print the <b><u>square of all the numbers</u></b> starting from <b>x</b> all the way up to <b>y</b>. After the program has reached this range, it will print “END”.

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>5 10</td><td>25 36 49 64 81 100 END</td></tr>
<tr><td>2 10</td><td>4 9 15 25 36 49 64 81 100 END</td></tr>
<tr><td>10 10</td><td>END</td></tr>
</table>
</td>
<td>**</td>
</tr>
<tr>
<td>6</td>
<td>
Write a program that takes a number, then print the reverse of that number.

[<b>Do this problem using the while statement</b>]

[<b>NOTE</b>: Do not use Python’s shortcuts. You must strictly use the loops.]

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>12345678</td><td>87654321</td></tr>
<tr><td>900</td><td>009</td></tr>
<tr><td>3</td><td>3</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td>7</td>
<td>
Write a program that will take the number <b>N</b> as input. Next, print the sum of first <b>Nth</b> terms for the following series:
1, -2, 3, -4, 5, -6, 7, -8, 9, -10, 11, -12, 13, -14, ...
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>2</td><td>3</td></tr>
<tr><td>10</td><td>11</td></tr>
<tr><td>0</td><td>1</td></tr>
</table>
<i>Explanation</i>

For the first example, the input is 2. Which means we need to find the 2nd index from the series. So 1 - (0th index), -2 - (1st index), 3 (2nd index)
</td>
<td>**</td>
</tr>
<tr>
<td>8</td>
<td>
You are working on a program to manage the roster of a sports team. The team's roster is represented as a list of jersey numbers, where each jersey number corresponds to a player. However, there seems to be an issue with duplicate jersey numbers, and you need to identify and resolve it.

Write a Python program that takes a list of jersey numbers as input and identifies and prints the jersey numbers that have duplicates. Your program should preserve the order of the original roster and only consider jersey numbers that have duplicates.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>3 7 2 8 2 5 7 1 8</td><td>Duplicate jersey numbers: 2 7 8</td></tr>
<tr><td>1 2 3 4 5</td><td>No duplicates</td></tr>
</table>
</td>
<td>**</td>
</tr>
<tr>
<td>9</td>
<td>
You are developing a salary calculation program for XYZ FinTech LLC. The company pays its employees <b>based on the number of hours worked and an hourly wage</b>. However, there are certain rules for overtime pay:
<ul>
<li>Regular hours (up to 40 hours): Paid at the regular hourly wage.</li>
<li>Overtime hours (more than 40 hours): Paid at 1.5 times <b>each hour</b> more than the regular hourly wage.</li>
</ul>
The program should then print the total salary for each employee.

The program takes two lists as input: one containing the hours worked by each employee and the other containing their corresponding hourly wage. Calculate the hourly wage of each employee in separate lines.

<i>Next page</i>
</td>
<td>**</td>
</tr>
</table>

Sample Input	**Sample Output**
Hours Worked: 38 45 30 50 Hourly Wage: 15 20 18 25	Employee 1: 570 BDT Employee 2: 950 BDT Employee 3: 540 BDT Employee 4: 1375 BDT

10

Design a program for your university's annual sports day, where traditional paper records are replaced by a sensor-based system to determine the winners of a 10m race. The program should accept participants' finish times as input, **with each index representing a participant number and the value at that index representing their finish time in seconds**. The program should then identify and display the participants who secured the first, second, and third positions. There can **never be two or more participants that finish at the same time**.

Sample Input	**Sample Output**
1 2 3 4 5	First Place: 0 Second Place: 1 Third Place: 2
5 1 2 4 9 10 7	First Place: 1 Second Place: 2 Third Place: 3
5 1 9	First Place: 1 Second Place: 0 Third Place: 2

Explanation:

For the second test case 5 1 2 4 9 10 7, the fastest time is 1 which is at index 1. So Participant 1 is first. The second fastest time is 2 at index 2, so Participant 2 is second. The third fastest time is 4 at index 3, so Participant 3 is third.

***

NEXT PAGE FOR THE SECOND SECTION

Section 2: Nested Loops, 2D Lists, and Patterns

<table>
<tr><th>SL.</th><th>Problem</th><th>Difficulty</th></tr>
<tr><td><b>11</b></td><td>Write a program to take a number N as input. Now, draw a right-angled triangle using *, up to the height N.
<b>[NOTE: Use loops]</b>
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>3</td><td>*
**
***</td></tr>
<tr><td>5</td><td>*
**
***
****
*****</td></tr>
<tr><td>100</td><td>(<b>Not an output.</b> You get the idea of how this should print)</td></tr>
</table></td><td>*</td></tr>
<tr><td><b>12</b></td><td>Write a program to take a number N as input. Now, draw a right-angled triangle that contains the squares of each row as their fill content (carefully check the output, print spaces between each fill content).
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>3</td><td>1
44
999</td></tr>
<tr><td>5</td><td>1
4 4
9 9 9
16 16 16 16
25 25 25 25 25</td></tr>
</table></td><td>*</td></tr>
</table>

<table>
<tr><td><b>13</b></td><td>Write a program to take a number N as input. Now, draw a hollow triangle using *.
<table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>3</td><td> *
 * *
*****</td></tr><tr><td>5</td><td>   *
   * *
  *   *
 *   *
*********</td></tr></table></td><td>**</td></tr>
<tr><td><b>14</b></td><td>Write a program to print to take a number N as input. The program should print this diamond pattern with numbers.
<table><tr><th>Sample Input</th><th>Sample Output</th></tr><tr><td>3</td><td> 1
 212
32123
 212
  1</td></tr><tr><td>5</td><td>   1
   212
  32123
 4321234
543212345
 4321234
  32123
   212
    1</td></tr></table></td><td>**</td></tr>
<tr><td><b>15</b></td><td></td><td></td></tr>
<tr><td><b>16</b></td><td></td><td></td></tr>
<tr><td><b>17</b></td><td></td><td></td></tr>
</table>

18		
19		
20		