

HW3: Mafia

HW3: Mafia

Release date: 03/11 16:00

Due date: 03/20 21:00

TA hour: 03/18 17:30-18:10

Backstory

Following the death of the mafia's big boss in a recent police raid, a power vacuum has thrown the organization into chaos, with members fighting among themselves due to old grudges and issues. Without a clear leader, conflicts arise as former allies turn into rivals.

As an undercover agent hiding in the organization, you have gathered all the needed information among all those mafia members. Your goal is to find out what will happen in the fight. The mafia members follow a very specific rule when it comes to fighting, it's called "tradition". The rules and other information you need to know are listed on the section below.

Description

- To follow the tradition, the members line up in a straight line.
- There are N members in the fight, with an index i denoting their position on the line.
 $i \in \{0, 1, \dots, N - 1\}$
- Each mafia member has two important traits: *Level* and *Range*. So there will be two sequences containig the level and range of the members according to their positions.
Level = $\{\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3 \dots \mathbf{L}_{N-1}\}$
Range = $\{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3 \dots \mathbf{R}_{N-1}\}$
- To determine whether a member at position i can attack a member at position j , it must satisfy the following three conditions:

1. $|j - i| \leq R_i$
member_j must be within the range of member_i's range .
 2. $L_j < L_i$
member_i must have higher level than member_j .
 3. $\{L_k\} < L_i$ for all $k \in \{k \mid i + 1 \leq k \leq j - 1\}$
member_i can't attack member_j if there's a member_k with higher level between member_i and member_j
- Here we denote that a_i and b_i will be the smallest index and the largest index the *member_i* can attack.
 - Please determine the sequence of pairs $\{(a_0, b_0), \dots, (a_{N-1}, b_{N-1})\}$

```
public static void main(String[] args) {  
    Mafia sol = new Mafia();  
    System.out.println(Arrays.toString(  
        sol.result(new int[] {11, 13, 11, 7, 15},  
                    new int[] { 1,  8,  1, 7,  2})));  
    // Output: [0, 0, 0, 3, 2, 3, 3, 3, 2, 4]  
    //      => [a0, b0, a1, b1, a2, b2, a3, b3, a4, b4]  
}
```

Level:	11	13	11	7	15
Range:	1	8	1	7	2

Index:

0

1

2

3

4



(0, 0)

(0, 3)

(2, 3)

(3, 3)

(2, 4)

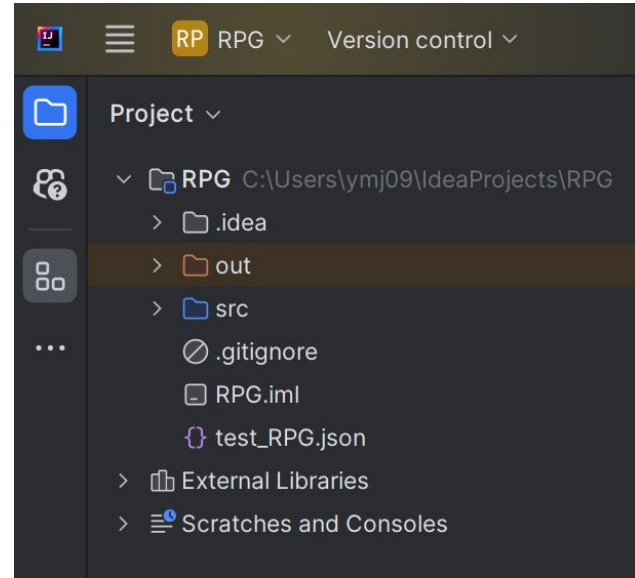
HOW TO USE THE TEST DATA (REVIEW!!!)

Tutorial For IntelliJ

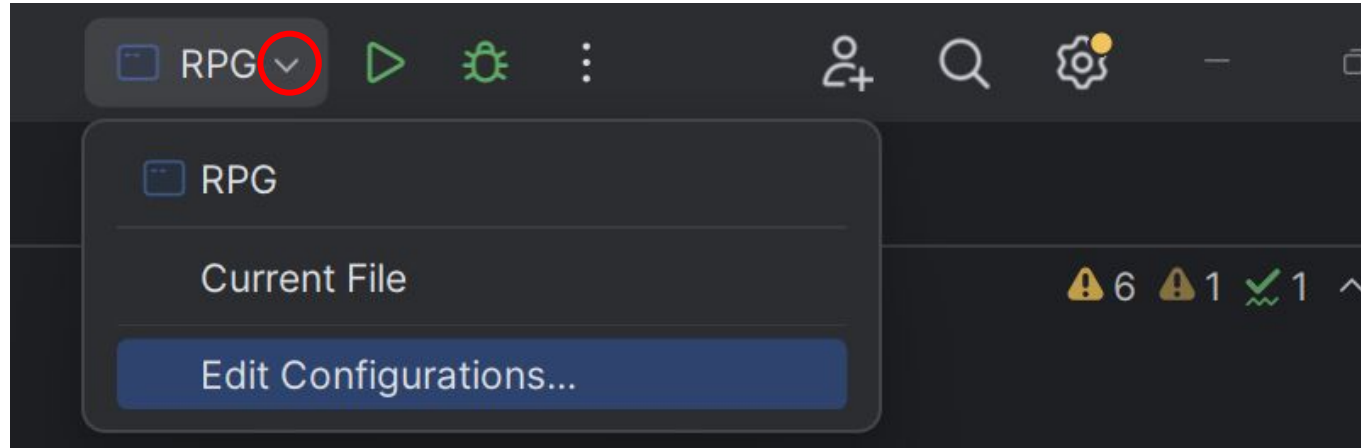
Paste the testcode and put the test data into the folder

```
© Main.java x
1 > /.../
36
37
38 import java.io.FileNotFoundException;
39 import java.io.FileReader;
40 import java.util.Arrays;
41
42 import com.google.gson.*;
43
44 3 usages
45 class OutputFormat{
46     2 usages
47     int[] defence;
48     2 usages
49     int[] attack;
50     1 usage
51     int k;
52     2 usages
53     int answer;
54 }
55
56 > class test_RPG{...}
57
58 > class RPG {...}
59
134
```

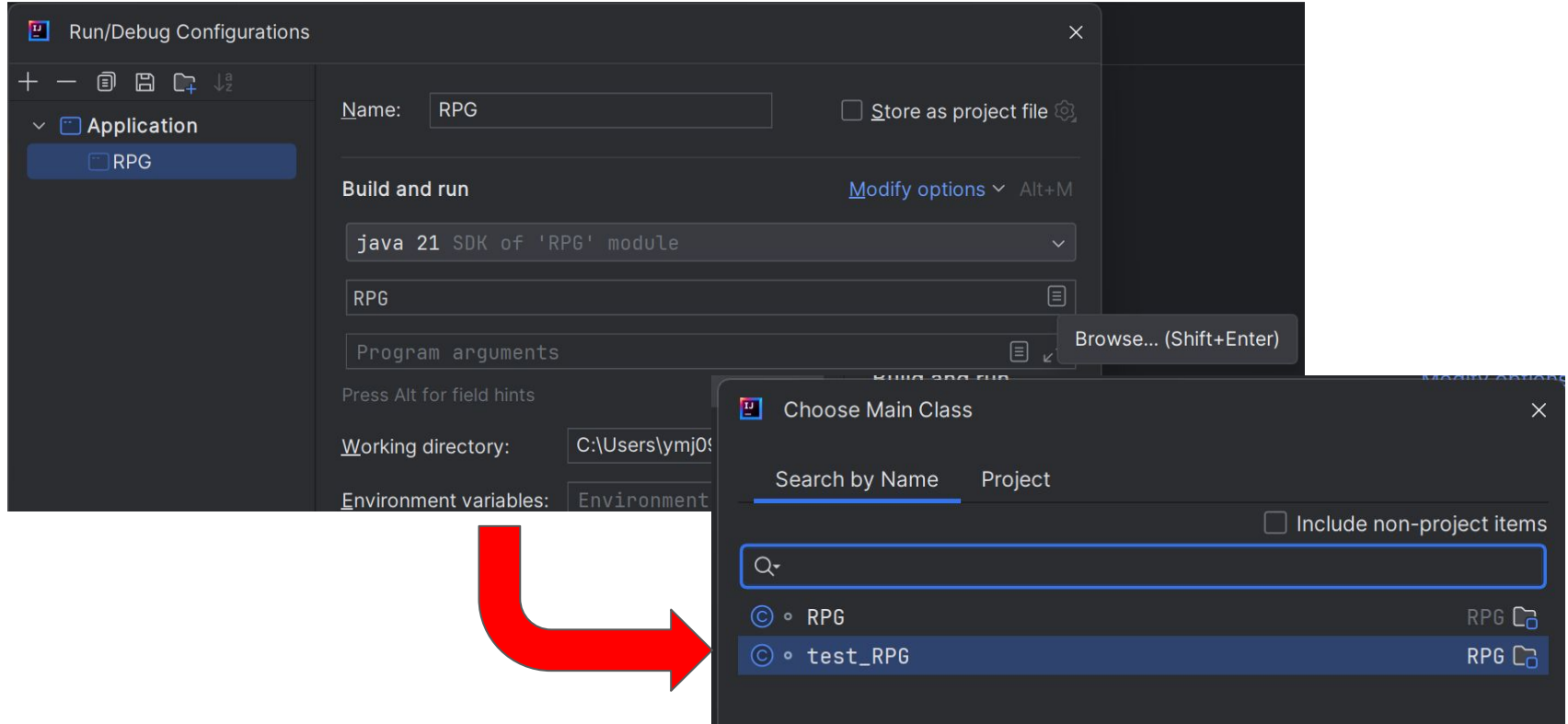
paste the whole test code above your own class, **no need to overwrite your own code.**



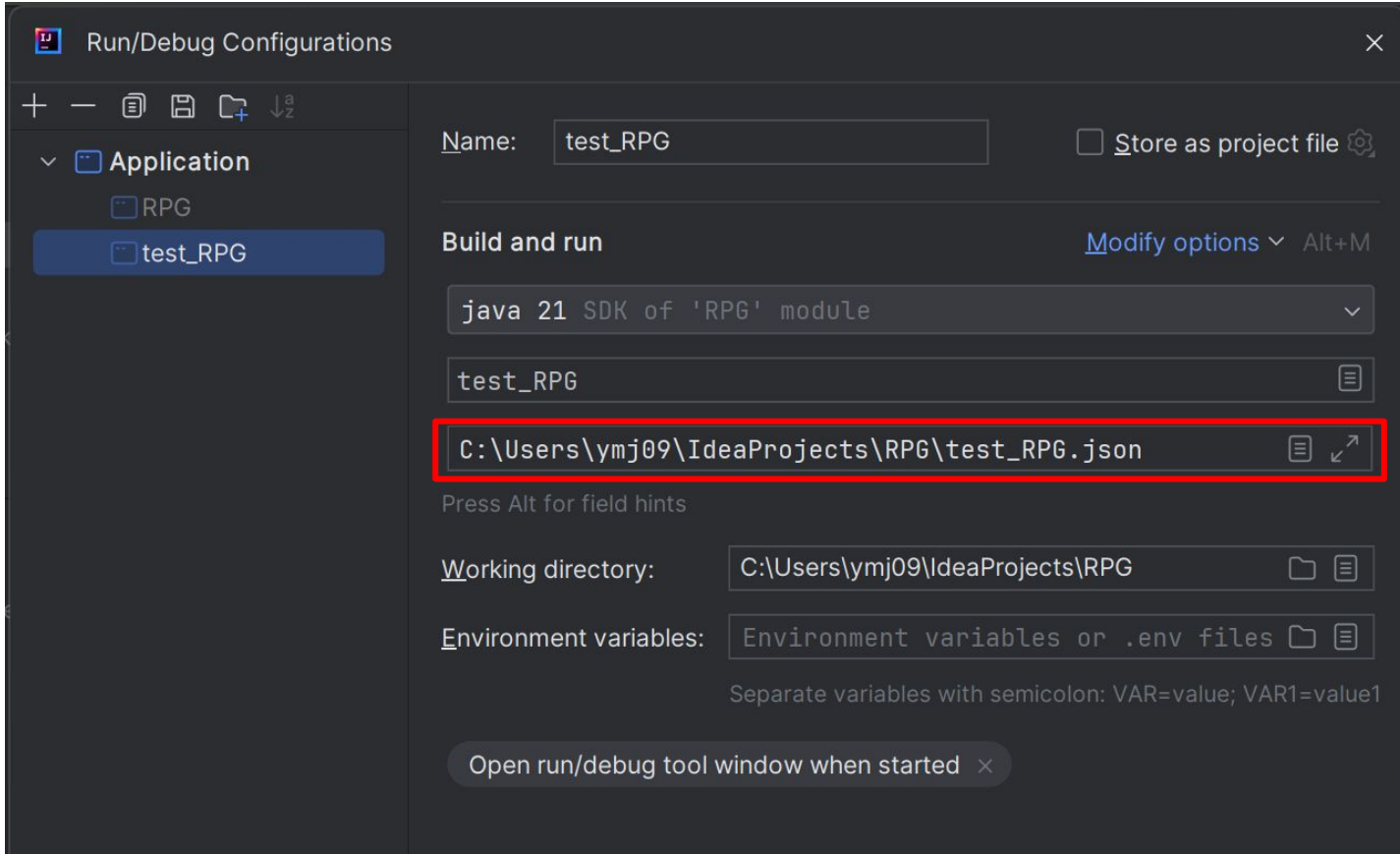
Click on the “V” beside the Run button and choose “Edit Configurations...”



Click on the button beside “RPG” and change the class you want to run to test_RPG



Paste the path of test_RPG.json here then you can run it



It should look like this if your code works properly~

```
C:\Users\ymj09\.jdk\openjdk-21.0.1\bin
Sample0: AC
Sample1: AC
Sample2: AC
Sample3: AC
Sample4: AC
Score: 5/5

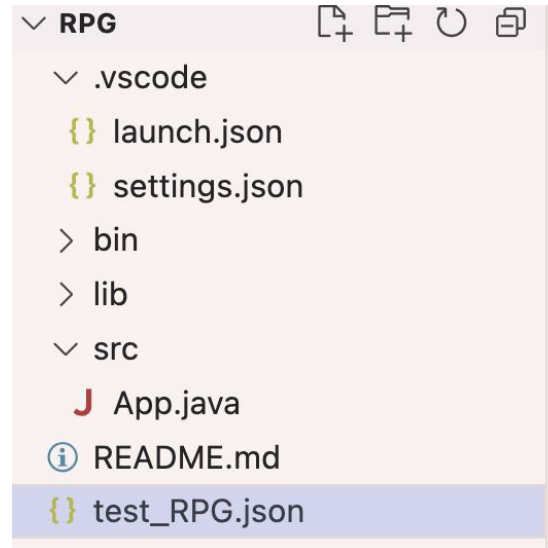
Process finished with exit code 0
```

Tutorial For VSCode

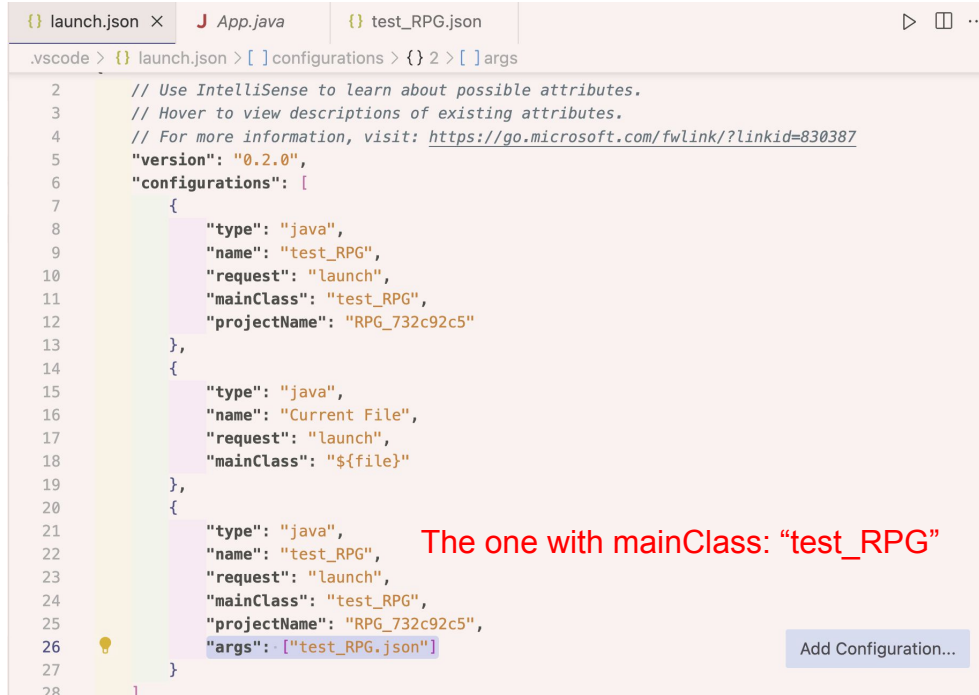
Paste the testcode and put the test data into the folder

```
src > J App.java > test_RPG > main(String[])
8   int[] attack;
9   int k;
10  int answer;
11  }
12
13  class test_RPG {
14      public static void main(String[] args) {
15          Gson gson = new Gson();
16          OutputFormat[] datas;
17          int num_ac = 0;
18          int user_ans;
19          OutputFormat data;
20
21          try {
22              datas = gson.fromJson(new FileReader(args[0]), OutputFormat[].class);
23              for (int i = 0; i < datas.length; ++i) {
24                  data = datas[i];
25                  user_ans = new RPG(data.defence, data.attack).maxDamage(data.k);
26                  System.out.print("Sample" + i + ": ");
27                  if (data.answer == user_ans) {
28                      System.out.println("AC");
29                      num_ac++;
30                  } else {
31                      System.out.println("WA");
32                      System.out.println("Data_atk: " + Arrays.toString(data.attack));
33                      System.out.println("Data_dfc: " + Arrays.toString(data.defence));
34                      System.out.println("Test_ans: " + data.answer);
35                      System.out.println("User_ans: " + user_ans);
36                      System.out.println("");
37                  }
38              }
39              System.out.println("Score: " + num_ac + "/" + datas.length);
40          } catch (JsonSyntaxException e) {
41              e.printStackTrace();
42          } catch (JsonIOException e) {
43              e.printStackTrace();
44          } catch (FileNotFoundException e) {
45              e.printStackTrace();
46          }
47      }
48  }
49
50  class RPG {
51      public RPG(int[] defence, int[] attack){
52          // Initialize some variables
```

paste the whole test code above your own class, **no need to overwrite your own code.**



Edit the launch.json in vscode



```
.vscode > {} launch.json > [ ] configurations > {} 2 > [ ] args
2 // Use IntelliSense to learn about possible attributes.
3 // Hover to view descriptions of existing attributes.
4 // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5 "version": "0.2.0",
6 "configurations": [
7   {
8     "type": "java",
9     "name": "test_RPG",
10    "request": "launch",
11    "mainClass": "test_RPG",
12    "projectName": "RPG_732c92c5"
13  },
14  {
15    "type": "java",
16    "name": "Current File",
17    "request": "launch",
18    "mainClass": "${file}"
19  },
20  {
21    "type": "java",
22    "name": "test_RPG",
23    "request": "launch",
24    "mainClass": "test_RPG",
25    "projectName": "RPG_732c92c5",
26    "args": ["test_RPG.json"]
27  }
28 ]
```

The one with mainClass: "test_RPG"

Add Configuration...

