
HW2: RPG

Backstory

You are playing a console game called "Dragon and Hero". You've finally reached the final boss battle. Your hero must beat up the Evil Dragon in order to win the game, but it's a big challenge. After failing countless times, You decided to find other ways to win the game (Cheating). You found a way to hack into the game's source code and discovered that the Evil Dragon's defense and the hero's attack was hardcoded into the program. The defense of the Evil Dragon and the hero's attack on each round doesn't change every time you replay the game. Therefore, you just have to find the perfect attack and boost combo with the knowledge you just learned from PDSA-2024-spring to win.

Description

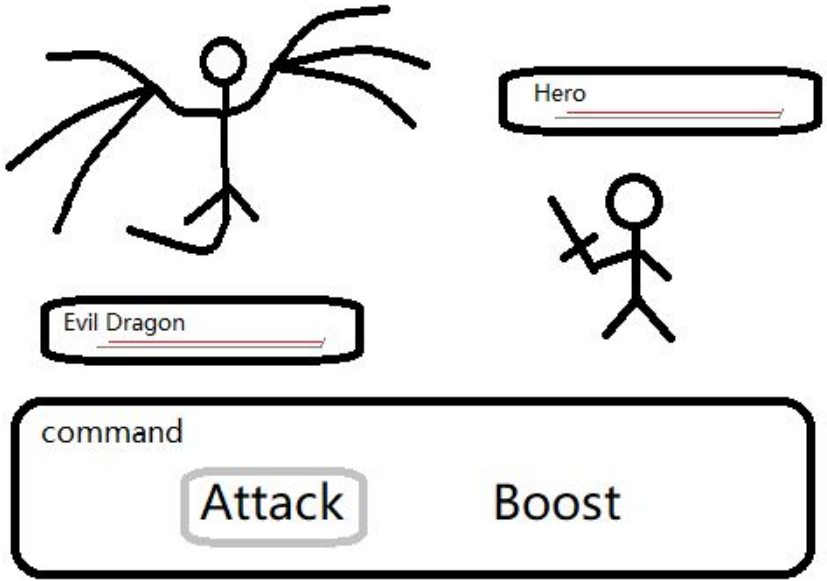
The game is round-based and for each round, the Hero can choose from one of the following actions:

Attack: **The Hero attacks the Dragon with a weapon.**

Boost: **Instead of attacking right away, the Hero prepares for a super-strong attack on the next round. The attack is doubled only for the next round.** Also, do remember that there's no damage done to the Dragon on the round the Hero chooses to boost. Since the boost only works for one round, the effect doesn't accumulate. (Please read the examples below)

Data of the game(Given as input)

- $\mathbf{Attack} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4 \ldots . \mathbf{a}_k \ldots . \mathbf{a}_N\}$
- $\mathbf{Defence} = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4 \ldots . \mathbf{d}_k \ldots . \mathbf{d}_N\}$



About Damage(What you need to find)

- $\mathbf{Damage} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4 \ldots . \mathbf{e}_k \ldots . \mathbf{e}_N\}$
- If you choose to attack:
- $\mathbf{Damage}_k = \mathbf{a}_k - \mathbf{d}_k$ **Normal attack - Normal defence**
- If you choose to boost:
- $\mathbf{Damage}_k = 0$, and $\mathbf{Damage}_{k+1} = \mathbf{2a}_{k+1} - \mathbf{d}_{k+1}$

You don't attack this round

Next round:
Boosted attack - Normal defence

Find the best combination of actions for the max damage

Specific task

For each round 1, 2, 3, and so on, we need to decide whether it's better to attack or boost. We want to figure out the best strategy that will give us the highest total damage after a specific number of rounds (let's call it n). Please return the highest total damage after n rounds.

$$\mathbf{maxDamage(n)} = max(\sum_{k=1}^{n} \mathbf{Damage}_k)$$

Examples

example1:

- attack: {235,234}
- defence: {100,20}
- k = 1

round 1: attack
Damage = (235 - 100) = 135

round 1: boost
Damage = 0

maxDamage(1) = 135
After round 1, the max amount of damage you can do to the dragon is going to be 135.

example2:

- attack: {235,234}
- defence: {100,20}
- k = 2

round 1: attack, round 2: attack
Damage = (235 - 100) + (234 - 20) = 349

round 1: boost, round 2: attack
Damage = 0 + (234*2 - 20) = 448

round 1: boost, round 2: boost
Damage = 0 + 0 = 0

round 1: attack, round 2: boost
Damage = (235 - 100) + 0 = 135

maxDamage(2) = 448
After round 2, the max amount of damage you can do to the dragon is going to be 448.

example3:

- attack: {235,234,200}
- defence: {100,20,150}
- k = 3

round 1: attack, round 2: attack, round 3: attack
Damage = (235 - 100) + (234 - 20) + (200 - 150) = 399

round 1: boost, round 2: attack, round 3: attack
Damage = 0 + (234*2 - 20) + (200 - 150) = 498

round 1: attack, round 2: boost, round 3: attack
Damage = (235 - 100) + 0 + (2*200 - 150) = 385

round 1: attack, round 2: attack, round 3: boost
Damage = (235 - 100) + (234 - 20) + 0 = 349

round 1: boost, round 2: boost, round 3: attack
Damage = 0 + 0 + (2*200 - 150) = 250
As you can tell, boosting for two consecutive rounds won't boost the attack of the 3rd round by 2^2.

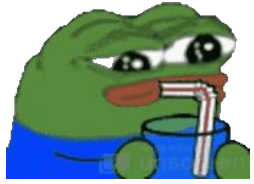
round 1: attack, round 2: boost, round 3: boost
Damage = (235 - 100) + 0 + 0 = 135

round 1: boost, round 2: boost, round 3: boost
Damage = 0 + 0 + 0 = 0

maxDamage(3) = 498
After round 3, the max amount of damage you can do to the dragon is going to be 498.

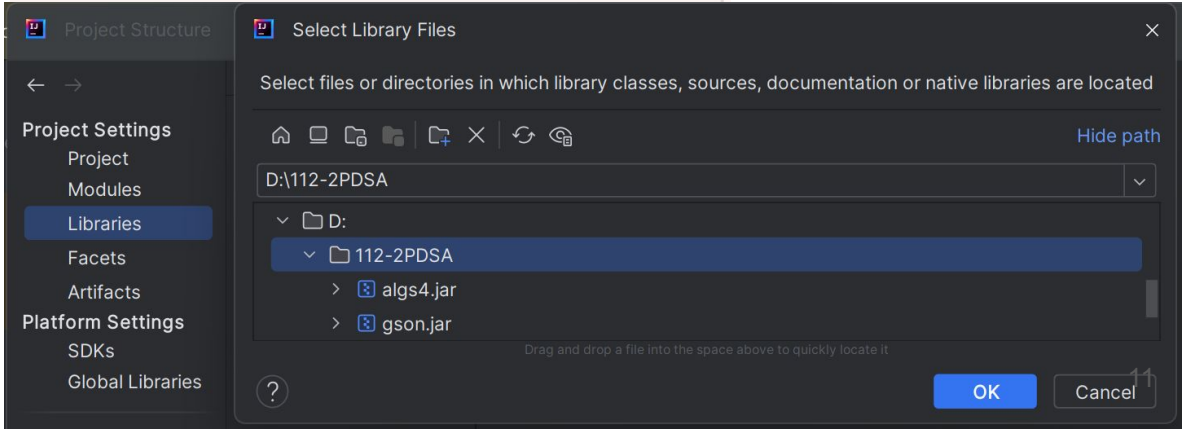
How to use the test dataset(Sorry it’s different from hw1)

Import both the libraries in the hackmd bottom description if you haven’t done so.(gson and algs4)



File Download

Data for TestCase
Test Code
algs4.jar library
gson.jar library



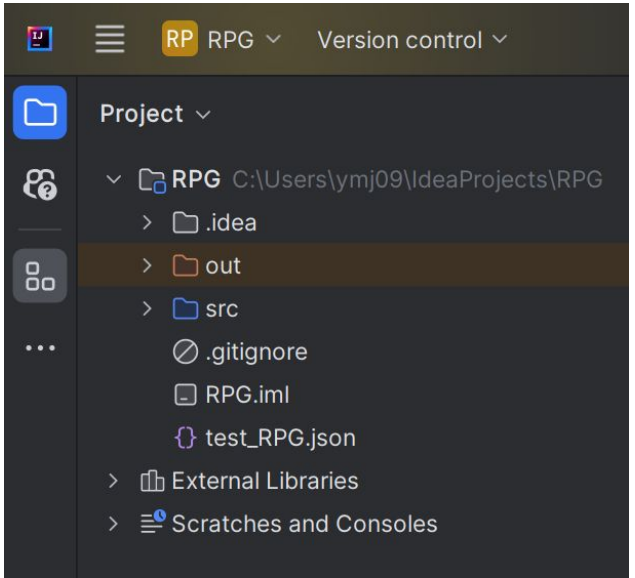
Tutorial For IntelliJ

Paste the testcode and put the test data into the folder

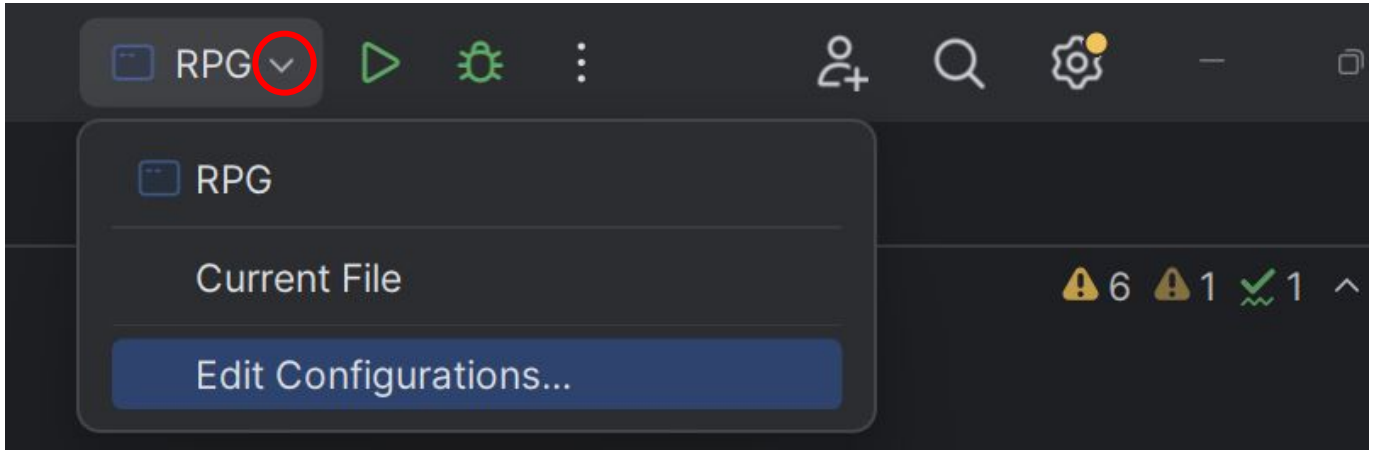
Main.java

```
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
    3 usages
44  class OutputFormat{
        2 usages
45      int[] defence;
        2 usages
46      int[] attack;
        1 usage
47      int k;
        2 usages
48      int answer;
49  }
50
51  class test_RPG{...}
92
93  class RPG {...}
134
```

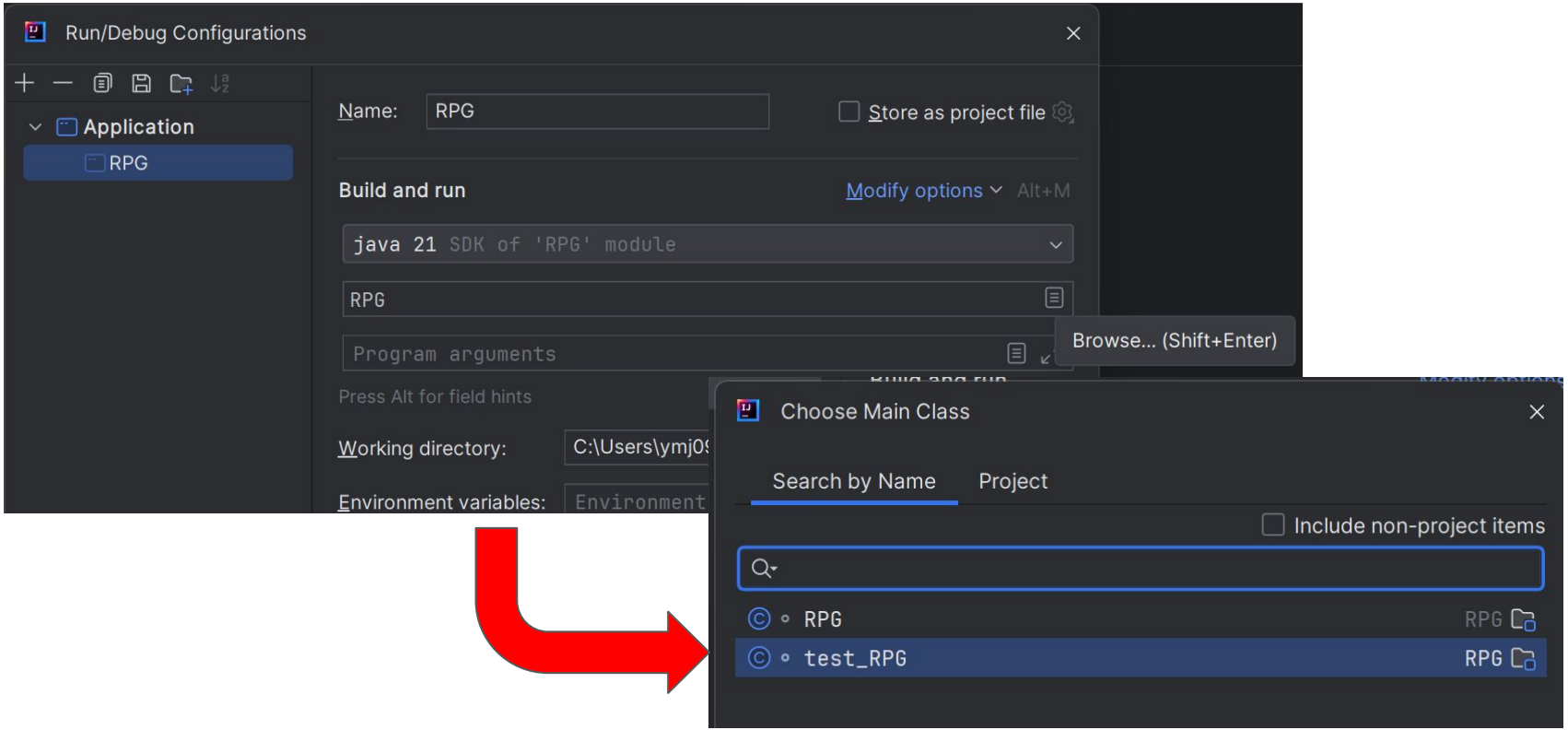
paste the whole test code above your own class, no need to overwrite your own code.



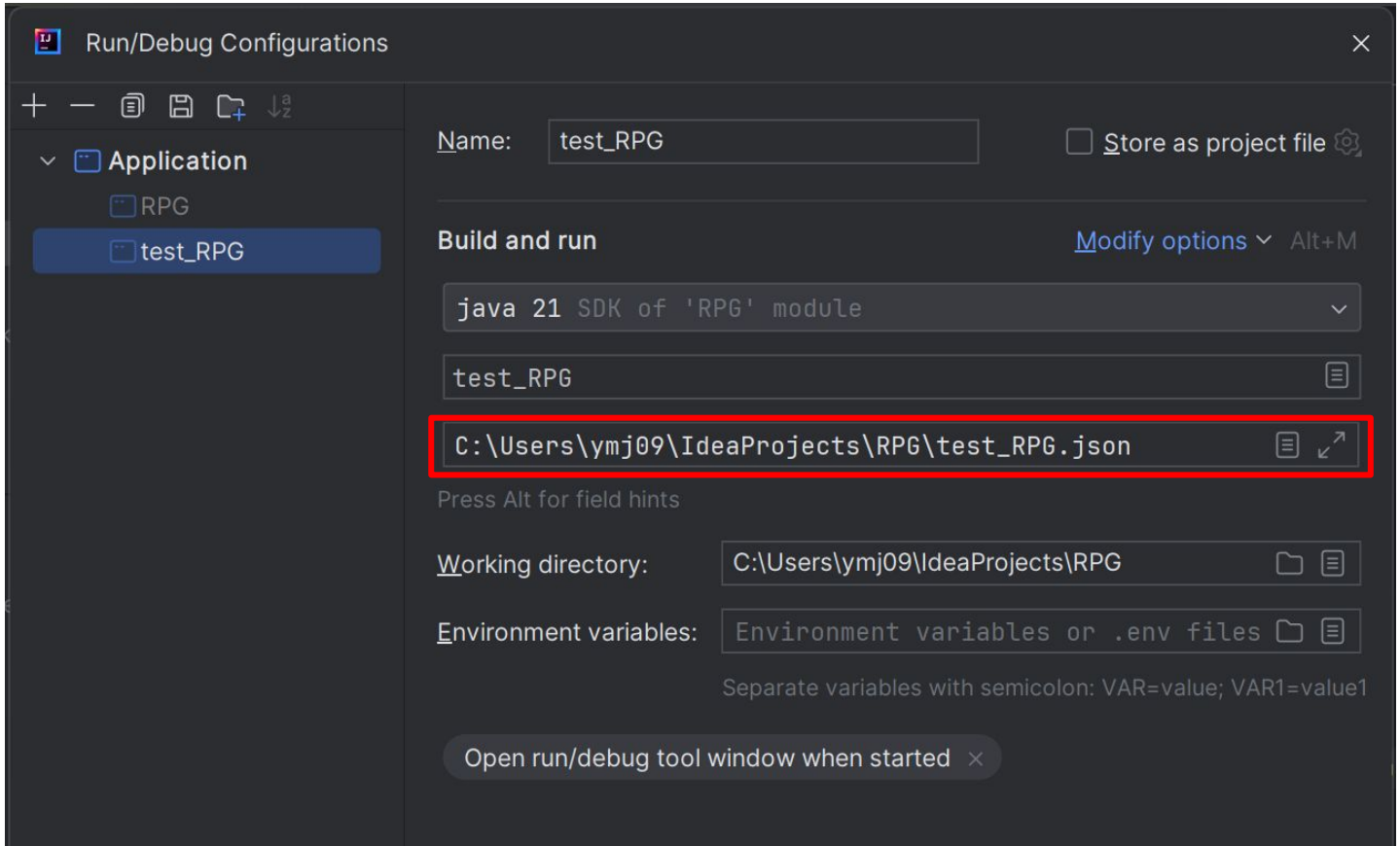
Click on the “V” beside the Run button and choose “Edit Configurations…”



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\.jdks\openjdk-21.0.1\b
Sample0: AC
Sample1: AC
Sample2: AC
Sample3: AC
Sample4: AC
Score: 5/5

Process finished with exit code 0
```

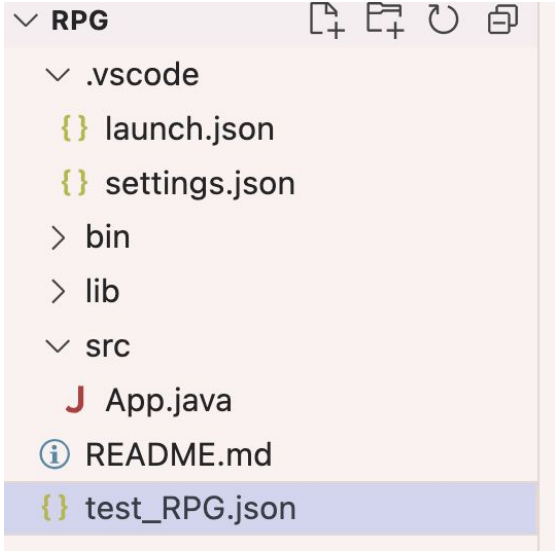
Tutorial For VSCode

Paste the testcode and put the test data into the folder

src > App.java > test_RPG > main(String[])

```
    int[] attack;
    int k;
    int answer;
}

class test_RPG {
    Run | Debug
    public static void main(String[] args) {
        Gson gson = new Gson();
        OutputFormat[] datas;
        int num_ac = 0;
        int user_ans;
        OutputFormat data;

        try {
            datas = gson.fromJson(new FileReader(args[0]), OutputFormat[].class);
            for (int i = 0; i < datas.length; ++i) {
                data = datas[i];
                user_ans = new RPG(data.defence, data.attack).maxDamage(data.k);
                System.out.print("Sample" + i + ": ");
                if (data.answer == user_ans) {
                    System.out.println(x:"AC");
                    num_ac++;
                } else {
                    System.out.println(x:"WA");
                    System.out.println("Data_atk:  " + Arrays.toString(data.attack));
                    System.out.println("Data_dfc:  " + Arrays.toString(data.defence));
                    System.out.println("Test_ans:  " + data.answer);
                    System.out.println("User_ans:  " + user_ans);
                    System.out.println(x:"");
                }
            }
            System.out.println("Score: " + num_ac + "/" + datas.length);
        } catch (JsonSyntaxException e) {
            e.printStackTrace();
        } catch (JsonIOException e) {
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

class RPG {
    public RPG(int[] defence, int[] attack){
```

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



Edit the launch.json in vscode

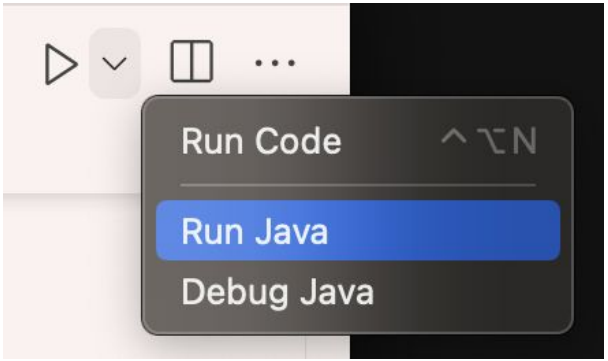
launch.json | App.java | test_RPG.json

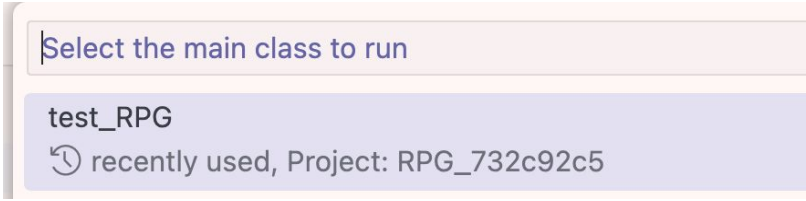
.vscode > launch.json > [ ] configurations > { } 2 > [ ] args

```
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
    {
        "type": "java",
        "name": "test_RPG",
        "request": "launch",
        "mainClass": "test_RPG",
        "projectName": "RPG_732c92c5"
    },
    {
        "type": "java",
        "name": "Current File",
        "request": "launch",
        "mainClass": "${file}"
    },
    {
        "type": "java",
        "name": "test_RPG",
        "request": "launch",
        "mainClass": "test_RPG",
        "projectName": "RPG_732c92c5",
        "args": ["test_RPG.json"]
    }
]
```

The one with mainClass: “test_RPG”

Add Configuration...





You should see this if things went well:)

```
Sample0: WA
Data_atk:  [263, 261, 264]
Data_dfc:  [90, 63, 119]
Test_ans:  604
User_ans:  0

Sample1: WA
Data_atk:  [224, 221, 222, 226, 223]
Data_dfc:  [168, 45, 195, 52, 88]
Test_ans:  424
User_ans:  0

Sample2: WA
Data_atk:  [240, 240, 245, 244, 247, 248, 247]
Data_dfc:  [127, 119, 142, 132, 170, 48, 12]
Test_ans:  717
User_ans:  0

Sample3: WA
Data_atk:  [279, 275, 274, 273, 272, 272, 276, 276, 275, 275]
Data_dfc:  [85, 68, 185, 118, 46, 146, 114, 145, 80, 69]
Test_ans:  194
User_ans:  0
```

If things don’t go expected, try to fix the launch.json if you can, or ask us in discord if you failed!