

## Group Assignment Topic

Semester 2, Session 2021/2022

## Topic D: World of Titan



Several hundred years ago, humanity was driven to the brink of extinction by the humanoid giants called [Titans](#), who apparently have no intelligence, and only attack and eat humans on sight. A small portion of humans retreated and formed a country behind extremely high walls, known as Paradis. Eren Yeager, a young man, suffered his childhood without his mother's love because she was eaten by a titan in one titan ambush incident. Since then, he decided to join the country's military forces and confront his fate to eliminate titans from the world.

### 1.0 Problem Statement

As Eren's friend who is familiar with the data structure knowledge, you need to help Eren solve the problems. So, these are the requirements:

- You can create a CLI based or GUI interface program.
- You are required to use the data structures and java to implement the features of your program.


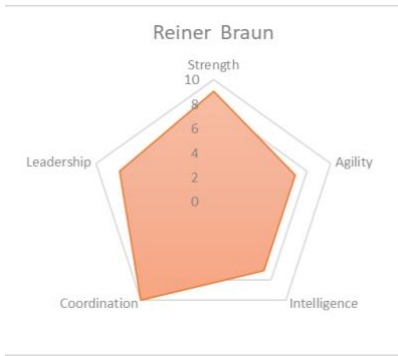

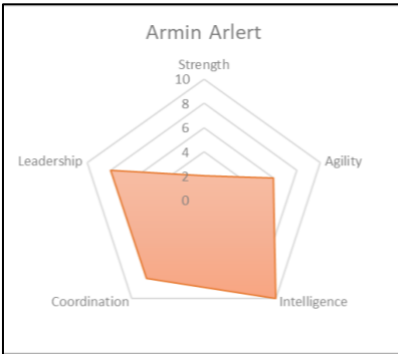
You can freely plan your solution, the images and the output provided below are only for reference purposes.


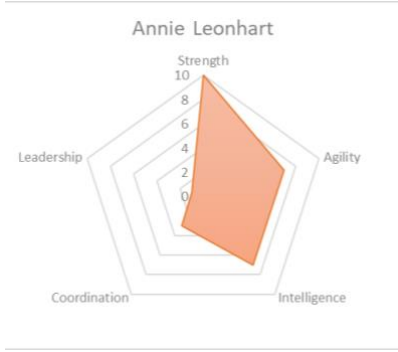

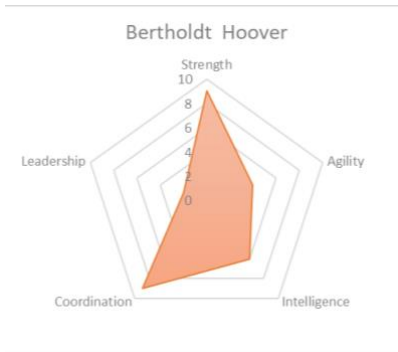

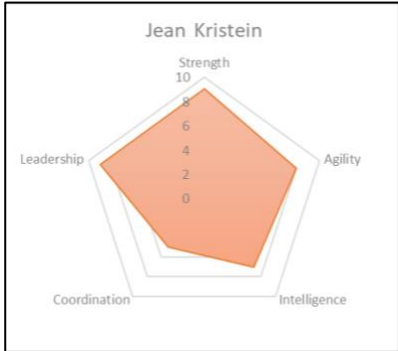

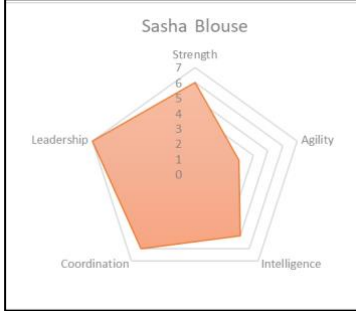
## 2.0 Basic Requirements


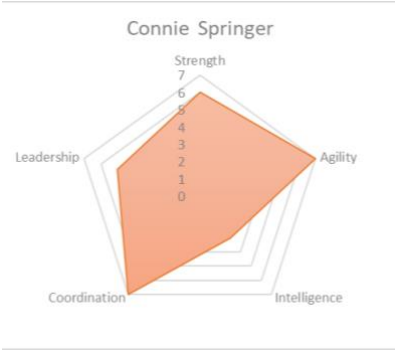
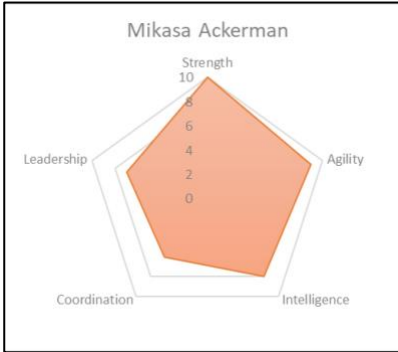

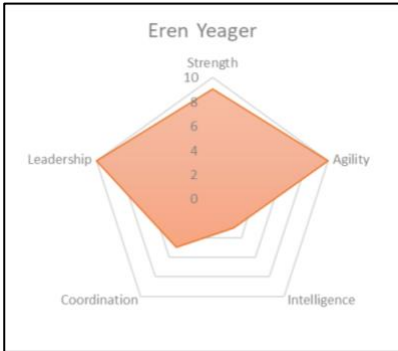

### 2.1 Eren's Allies






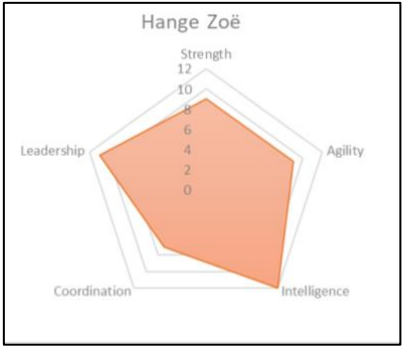
Eren has too many friends in the military, and Eren always forgets what the characteristics of their friends are. But Eren has provided you with a list that needs to be recorded in character class. The list below is from the note provided by Eren. You will need to store all the following characters listed in Table 1 below in the **linked list**.

Table 1: Eren's friends list.

Name	Characteristic	Characteristic in Graph View
Reiner Braun 	<ul style="list-style-type: none"> <li>Height: 185cm</li> <li>Weight: 95kg</li> <li>Strength: 9</li> <li>Agility: 7</li> <li>Intelligence: 7</li> <li>Coordination: 10</li> <li>Leadership: 8</li> </ul>	
Armin Arlert 	<ul style="list-style-type: none"> <li>Height: 163cm</li> <li>Weight: 55kg</li> <li>Strength: 2</li> <li>Agility: 6</li> <li>Intelligence: 10</li> <li>Coordination: 8</li> <li>Leadership: 8</li> </ul>	

<p>Annie Leonhart</p> 	<ul style="list-style-type: none"> <li>○ Height: 153cm</li> <li>○ Weight: 54kg</li> <li>○ Strength: 10</li> <li>○ Agility: 7</li> <li>○ Intelligence: 7</li> <li>○ Coordination: 3</li> <li>○ Leadership: 1</li> </ul>	
<p>Bertholdt Hoover</p> 	<ul style="list-style-type: none"> <li>○ Height: 192cm</li> <li>○ Weight: 81kg</li> <li>○ Strength: 9</li> <li>○ Agility: 4</li> <li>○ Intelligence: 6</li> <li>○ Coordination: 1</li> <li>○ Leadership: 2</li> </ul>	
<p>Jean Krstein</p> 	<ul style="list-style-type: none"> <li>○ Height: 175cm</li> <li>○ Weight: 65kg</li> <li>○ Strength: 9</li> <li>○ Agility: 8</li> <li>○ Intelligence: 7</li> <li>○ Coordination: 5</li> <li>○ Leadership: 9</li> </ul>	
<p>Sasha Blouse</p> 	<ul style="list-style-type: none"> <li>○ Height: 165cm</li> <li>○ Weight: 55kg</li> <li>○ Strength: 6</li> <li>○ Agility: 3</li> <li>○ Intelligence: 5</li> <li>○ Coordination: 6</li> <li>○ Leadership: 7</li> </ul>	

<p>Connie Springer</p> 	<ul style="list-style-type: none"> <li>○ Height: 158cm</li> <li>○ Weight: 58kg</li> <li>○ Strength: 6</li> <li>○ Agility: 7</li> <li>○ Intelligence: 3</li> <li>○ Coordination: 7</li> <li>○ Leadership: 5</li> </ul>	
<p>Mikasa Ackerman</p> 	<ul style="list-style-type: none"> <li>○ Height: 170kg</li> <li>○ Weight: 68kg</li> <li>○ Strength: 10</li> <li>○ Agility: 9</li> <li>○ Intelligence: 8</li> <li>○ Coordination: 6</li> <li>○ Leadership: 7</li> </ul>	
<p>Eren Yeager</p> 	<ul style="list-style-type: none"> <li>○ Height: 170cm:</li> <li>○ Weight: 63kg</li> <li>○ Strength: 9</li> <li>○ Agility: 10</li> <li>○ Intelligence: 3</li> <li>○ Coordination: 5</li> <li>○ Leadership: 10</li> </ul>	
<p>Historia Reiss</p> 	<ul style="list-style-type: none"> <li>○ Height: 145cm</li> <li>○ Weight: 42kg</li> <li>○ Strength: 4</li> <li>○ Agility: 8</li> <li>○ Intelligence: 7</li> <li>○ Coordination: 6</li> <li>○ Leadership: 5</li> </ul>	

<p>Levi Ackerman</p> 	<ul style="list-style-type: none"> <li>○ Height: 160cm</li> <li>○ Weight: 65kg</li> <li>○ Strength: 12</li> <li>○ Agility: 12</li> <li>○ Intelligence: 7</li> <li>○ Coordination: 8</li> <li>○ Leadership: 8</li> </ul>	
<p>Erwin Smith</p> 	<ul style="list-style-type: none"> <li>○ Height: 188cm</li> <li>○ Weight: 92kg</li> <li>○ Strength: 8</li> <li>○ Agility: 8</li> <li>○ Intelligence: 11</li> <li>○ Coordination: 10</li> <li>○ Leadership: 12</li> </ul>	
<p>Hange Zoë</p> 	<ul style="list-style-type: none"> <li>○ Height: 170cm</li> <li>○ Weight: 60kg</li> <li>○ Strength: 9</li> <li>○ Agility: 9</li> <li>○ Intelligence: 12</li> <li>○ Coordination: 7</li> <li>○ Leadership: 11</li> </ul>	

```
Enter name: Levi Ackerman
Enter characteristics: 160 65 12 12 7 8 8

Name: Levi Ackerman
Height: 160cm
Weight: 65kg
Strength: 12
Agility: 12
Intelligence: 7
Coordination: 8
Leadership: 8
```

Figure 1: Sample output for entering the character information from the console.

With referring to Figure 1, your program should receive character's name as first input, characteristics as second input and output the information entered. However, it is highly recommended to store the information in any file format (e.g., txt, docx, json) and load them into the program whenever the program starts so that you do not need to re-enter the information every time the program reboots.

## 2.2 Soldiers Arrangement and Grouping

Captain Erwin always has a hard time arranging the soldiers. Captain Erwin wants to put the soldiers in a certain order all the time. Use any kind of sorting method to sort the members based on the different types of attributes of the members, such as height, weight, and ability.

```
Sorting attribute: Strength

Levi Ackerman 12
Annie Leonhart 10
Mikasa Ackerman 10
Reiner Braun 9
Bertholdt Hoover 9
Jean Kristein 9
Eren Yeager 9
Hange Zoë 9
Erwin Smith 8
Sasha Blouse 6
Connie Springer 6
Historia Reiss 4
Armin Arlert 2
```

Figure 2: Sample output to sort the soldier according to strength. Your program should receive sorting attribute(s) as input, rank the soldiers based on the sorting attribute(s) and output the soldiers and their sorting attribute(s) based on the rankings.

To carry out the investigating action outside the Wall, Captain Erwin needs to find soldiers by using ability value to make team forming easier. To make finding more efficient, **binary search** is encouraged to help Captain Erwin in finding the soldier.

```
Finding ability : Strength
value: 10

Soldier : Annie Leonhart, Mikasa Ackerman
```

Figure 3: Sample output to find the soldier with strength with ability 10. The output is Annie Leonhart and Mikasa Ackerman.

## 2.3 Titan Evaluation and Killing Priority

Eren Yeager and his team had noticed that there were many different danger levels of the Titan. Titans with some features will be more dangerous and they will make more people die. These titans also have strong ability in fighting with the soldiers. This is how to evaluate the dangers of titans:

If Titan type is **normal** (sum all the danger risk to get the total danger risk):

- Height
  - height > 20m - danger risk 3
  - height > 10m - danger risk 2
  - height < 10m - danger risk 1
- Walking legs
  - 4 legs - danger risk 3
  - 2 legs - danger risk 2
  - 0 legs - danger risk 1
- Running speed
  - Speed > 20ms - danger risk 3
  - Speed > 10ms - danger risk 2



- Speed < 10ms - danger risk 1
- Walking pattern
  - Not repeated pattern - danger risk 3
  - Repeated pattern - danger risk 2
  - Normal pattern - danger risk 1
- Climbing skill
  - Can climb - danger risk 3
  - Cannot climb - danger risk 1

If Titan type is **abnormal**:

- Danger risk 15

If Titan type is one of the [nine titan](#):

- Danger risk 19

To ensure the safety of the residents, you should choose to **kill Titans with the greatest danger risk first**. Construct a **priority queue** to put the Titans' details and compute which Titan should be killed first. Calculate the distance moved in terms of the difference in index position of the titans in the input. Based on Figure 3, assume you are in index 0 and the first Titan will be located at index 1, second Titan located at index 2, etc. In this case, you will kill Titan 2 at index 2, followed by Titan 1 at index 1 and finally Titan 3 at index 3. The total distance moved will be from index 0 to index 2, index 2 to index 1 and index 1 to index 3 ( $2+1+2$ ) = 5.



```
Number of Titans: 3

Generating 3 Titans ....
Titan 1: Normal Titan (25m, 4 legs, 15ms, Repeated pattern, Can climb) Risk=13
Titan 2: Abnormal Titan Risk=19
Titan 3: Normal Titan (8m, 2 legs, 3ms, Normal pattern, Can climb) Risk=8

Sequence to be killed: Titan 2 --> Titan 1 --> Titan 3
Total distance moved: 5
```

Figure 4: Sample output for Titan killing priority. Your program should receive number of Titans as input, randomly generate Titans based on input and display kill sequence and total distance moved.

### 2.3 Scouting Mission inside the Wall

Captain Erwin gets the information from the garrison which there are some titans invading the wall (noticed that those titans are summoned by Beast Titan). After walking through the wall, survey corps found that there is no difference on the Wall. But, to clear all the titans inside that area. Scouting mission should be carried out to notice all titans' position.

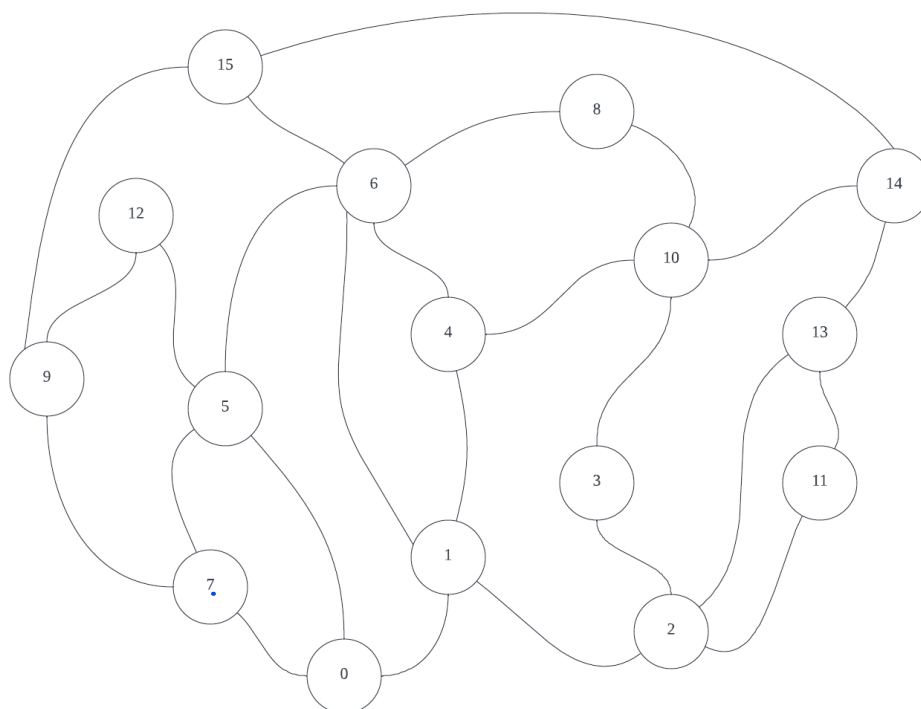


Figure 5: Map nodes in Paradis which the soldiers move using moving machine.

Find out one point that can search all the points **without any repeating on the points**. Then, getting back to that starting point (**Hamiltonian cycle**). Print all the node travel and the starting point you choose.

```
Enter starting point: 3
No path found.
Enter starting point: 6
Path found!
6-->8-->10-->3-->2-->11-->13-->14-->15-->9-->12-->5-->7-->0-->1-->6
```

Figure 6: Sample output for path finding. If you choose node 3 as the starting point, there is no Hamiltonian cycle so no path is displayed. If you choose node 6 as the starting point, there is a Hamiltonian cycle so the path (node travel) will be displayed.

## 2.4 Best Path to Kill Titan

There are too many barriers in the town when the titan enters Maria's wall. Many of the soldiers die because they use the long path to kill the titan. This causes the gas of their movement machine to become empty and get killed by the titan. So, to increase the survival rate of the soldiers, Eren has provided the map of the barriers in the town and the position of the titan, which will randomly appear in all the corners of the city. Below is a map of the city.

The titan will be generated at random in the graph node, excluding the node start. Given that the gas of the movement machine is used constantly with the passage of time, and all the time used to go to another node using the movement machine is only 1. Implement the algorithm to find the shortest path to kill the titan.

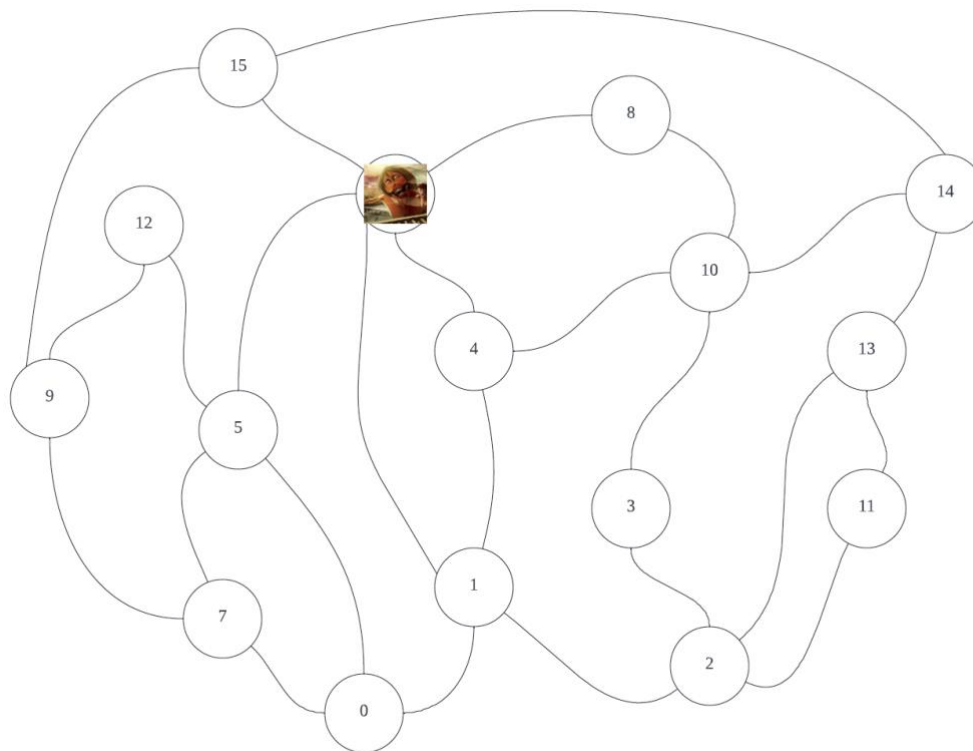


Figure 7: Map nodes in Paradis and Titan is on node 6

In this example shown in Figure 7, The titan is on the node 6. Then, the fastest path is [0,1,6] and [0,5,6] instead of [0,1,4,6]. Note that the starting node will always be 0 for this question.

```
Enter location of Titan: 6
```

```
Best paths:
```

```
0-->1-->6
```

```
0-->5-->6
```

Figure 8: Sample output for finding the best path. Your program should receive location of Titan and display the best path(s).

## 2.5 Marley word converter

After that, you tried hard in the war to get back Maria Wall. Eren had finally returned to his home to open the door to the basement in his house. Eren successfully gets the information from his father, Grisha. But all the information is in Marley's sentences. However, Eren found a dictionary in the book rack that is a Marley translation-based

dictionary. After Eren's study, he found that there was a pattern and relationship between the Paradis language and the Marley language. Eren hopes to achieve automated translation of the word to make sure he can get the word easily. So, Figure 8 shows the character translation provided by Eren and Figure 9 provides the grammar symbols in Marley word.

a	b	c	d	e	f	g	h	i	j	k	l	m
j	c	t	a	k	z	s	i	w	x	o	n	g

n	o	p	q	r	s	t	u	v	w	x	y	z
b	f	h	l	d	e	y	m	v	u	p	q	r

Figure 9: Marley character to Paradis character.

^	Character after the word will be in big letter
\$	This will be space in Marley character
,	This will be “,” in Marley character
()	The word inside parentheses will be inverted

Figure 10: Grammar in Marley word.

You are **prohibited from using the Java.Util.HashMap class** to solve the problem. You are required to implement your **own HashMap** to conduct the dictionary above and try your input given below.

Use of external libraries is prohibited for implementations of the basic requirements of this question. They are allowed for implementations of the additional challenges to not limit your creativity with what you can do. Please use them as a tool in your implementation and not the implementation itself. External libraries here are defined as libraries outside of the Java standard library.

```

Enter Marley sentence: oh(ldchc$ebdccc$rl)
find attack titan
Enter Marley sentence: rsgc(qqd^i$tkz)$ko$^udzhd,(rld$sgk^z$)$^gpssld
destroy Wall of Maria, Rose and Sheena
Enter Marley sentence: ^ukgc$rd(vsq$gh$zshrqkg$gwksml)h$dbeszudl
Most dangerous soldier is levi ackerman

```

Figure 11: Sample output for automatic translation. Your program must be able to receive all Marley words in Figure 10 and convert them into Paradis Word.

## 2.6 Protecting Wall of Maria

Captain Erwin captures information that the Armored Titan and Colossal Titan will try to break the Wall of Maria again. They also knew their enemy had obtained the wall building structure of Wall of Maria, and they will break the weakest part of the wall.

Wall of Maria is built of different blocks and layers, you will be given the structure of Wall of Maria as a `List<List<Integer>>`, which presents how Wall of Maria's structure. The weakest part of the wall is considered a vertical line that cuts through the least number of wall layers, and goes through most edges of bricks.

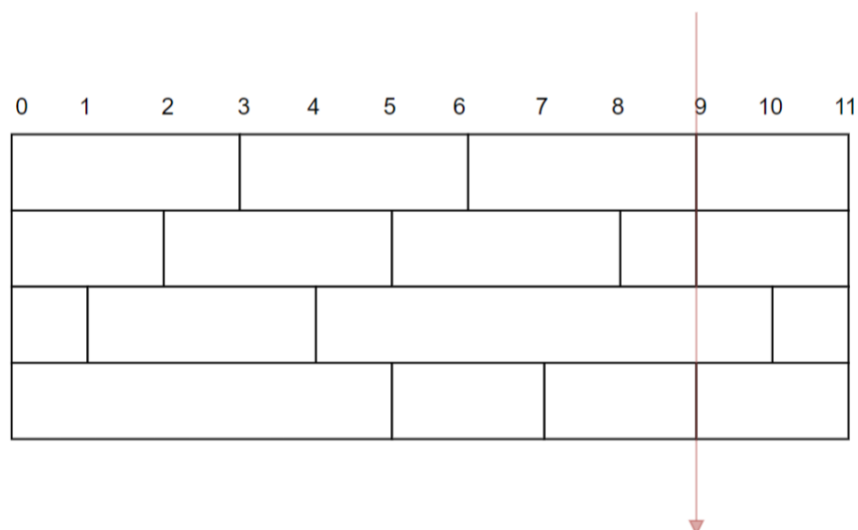


Figure 12: Example of Wall of Maria structure and its weakest part.

From figure above, that weakest part of the wall will be at index 9, which only cuts through one wall. Help Captain Erwin to find the weakest part of the wall so that he can assign soldiers to guard the part of the wall.

```
Enter number of layers: 4
Enter brick edges of layer 1: 3 6 9
Enter brick edges of layer 2: 2 5 8 9
Enter brick edges of layer 3: 1 4 10
Enter brick edges of layer 4: 5 7 9

Weakest part of the wall is at position 9
```

Figure 12: Sample output for finding the weakest part of the wall.

### 3 Additional Challenges / Extra Features

This section requires you to implement things that might be beyond your learning syllabus. Hence, you're free to choose any of them to implement it in your program. Completing the extra feature(s) will grant you extra mark(s), and make your program stand out among your peers. You're encouraged to try more than one, if possible, to maximize your learning outcome.

#### I. GUI

Build your Attack on Titan military system with a nice-looking GUI. It is possible to simulate it in a graph or in a graph with a graphic map background. Your program should simulate the process of killing the titan (movement of soldiers to the titan with respect to time).

#### II. Extra Algorithm Implementations

You can implement other searching algorithms to search for a best-known path for the titan killing process. Here are some of the possible searching algorithms you might want to consider:

- a) Best First Search
- b) A\* Search
- c) Genetic Algorithm
- d) Dijkstra algorithm
- e) Your custom searching algorithm

### III. Titan evaluation and killing priority

One of the soldiers is chosen to kill the titans there. The worst situation happened there, because some new dangerous titans were added to the queue at the halfway point. You are required to use data structure knowledge to compute the queue checking and detect the new most dangerous titan. Sometimes, soldiers should run if they don't have the ability to kill a dangerous titan. Noticed that the strength and agility of the character are important to killing the dangerous titan. If the sum of strength and agility is greater than or equal to the danger risk, then process the calculation of the distance operation. Else, the soldier will be dead if you do not run.

### IV. Best path to kill Titan 1

Titan is not stationary, and they will move over time. Given the path of the titan, find out the shortest path to kill the titan and which node you will kill the titan. The titan will move for each 2 times, to another node. If their path is in [1,3,5], if the time is 2 its location will be node 3, if the time is 4, its location will be in node 5. Find out the shortest path

### V. Best path to kill Titan 2

The Beast Titan comes to Paradis and kills many soldiers. Its ability is to summon the titan and throw the stones. This has damaged the soldiers of Paradise seriously. It has been protected by building nodes, titan nodes and tree nodes. Beast Titan is located at one of the nodes, and finds the shortest path to kill the Beast Titan. Find the shortest path. But, you will have other time differences based on the chosen character during passing building nodes, titan nodes, and tree nodes.

If coordination < 5 pass building, time = 3, else if coordination < 8, time = 2, else time = 1;

If intelligence < 5 pass titan, time = 3, else if intelligence < 8, time = 2, else time = 1;

If agility < 5, pass tree, time = 3, else if agility < 8, time = 2, else time = 1;



Given node example:

1, 3, 6, 8, 10, 11, 13 are building node

2, 12, 14 are titan node

4, 5, 7, 9, 15 are tree node

Choose a character to use its ability to find the most suitable path for them.

## VI. Real Battle

Titans with danger risk in 3th problem will be generated in the graph. Kill them based on priority and return distance travel. If the titan is too dangerous, the soldier should run, find out the shortest path to kill all titans. Note that the titans also will enter the battle halfway.

## VII. Marley word converter with cipher

To prevent being exposed, imposters in Paradis try to use the code method to encrypt their messages. But, Captain Erwin got the information on one occasion. He finds Eren to help. Eren notices that some type of encryption method is used in the Marley word. Carsar's cypher and changing the place of characters in sentences are used.

&num{}	All of the character should be subtracted with num before decrypt it
--------	--

Find out who is the imposter. Also, you are required to create your own cipher algorithm. You may refer to some [common cipher algorithm](#) that work. Explain how secure your cipher is.

## VIII. Always Think Out of the Box

You are always encouraged to brainstorm and try out anything.

## 4.0 Others

### 4.1 Marking

Kindly refer to the assignment instruction and marking rubrics for more information.

### 4.2 Tips

Here are some tips to help you in completing this assignment:

- Always plan ahead in terms of work distribution among team members. As you have already known, this is a group assignment, hence you're expected to work in a team and present the outcome together.
- Use code tracking tools such as Git or SVN to better align the codes among your team members.
- Give meaningful comments in your code. This will help the readers (your team members and lecturer) to understand your logic easier and faster, reducing the amount of time needed to figure out the whole picture.

### 4.3 References

If you wish to understand more about Attack on Titan to complete the assignment, I highly recommend you refer to this [resource link](#).

### 4.4 Enquiries

If you have any questions regarding the question, you can email to [erenyeager.ds@gmail.com](mailto:erenyeager.ds@gmail.com).

### end of assignment topic description ###

## good luck and have fun! ##