VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMMING FUNDAMENTALS - CO1027

ASSIGNMENT 1

PRINCESS GUINEVERE

Version 1.0

Ho Chi Minh City, 02/2023

ASSIGNMENT'S SPECIFICATION

Version 1.0

1 Assignment's outcome

After completing this assignment, students review and make good use of:

- Conditional statements
- Loop statements
- Array and 2-dimensional array
- String processing
- Function and function call
- File read/write operations

2 Introduction

Princess Guinevere, daughter of King Leodegrance, was a beautiful lady, the betrothed wife of the divine King Arthur. Rumors of her beauty had crossed the borders of England, reaching the ears of the ferocious dragon Bowser that reigned in the distant kingdom of Koopa. Bowser kidnaps Guinevere and brings her back to Koopa to force her to be his wife.

At this point, Arthur had just defeated Cedric and ascended the throne of England. Hearing that his lover was kidnapped, King Arthur first wanted to go and rescue Guinevere. But leaving England at this time would mean once again the kingdom fell into chaos and chaos, the stability and peace that Arthur had worked so hard to build could very well be dissipated.

The Knights of the Round Table, led by the knight Lancelot, asked King Arthur to continue stabilizing England's interior. The rescue of Princess Guinevere will be handled by the Knights of the Round Table. Whether King Arthur will abandon his kingdom for love, whether the Knights of the Round Table can defeat the formidable dragon Bowser to rescue Princess Guinevere, all will be answered in this great exercise.

In this assignment, students will be provided with a file containing input data, including information about the knight of the Round Table who rescued Guinevere and data about the journey to the kingdom of Koopa. The program will print to the screen the results of the Guinevere rescue process (more detailed in the section 4).

3 Input data

The program's input is contained in a file, the filename will be passed in when the program is run. This file will contain the following information:

- Information about the Knight of the Round Table, for example, is as follows:

```
172␣2␣0␣1␣1
```

- The itinerary to Koopa, for example, is as follows:

```
1␣2␣9␣18␣99
```

- Additional data file names, for example, are as follows:

```
mush_ghost,,merlin_pack
```

Thus the input file will consist of 3 lines. The first line will describe the information of the knight participating in the rescue of Guinevere in the following format:

```
HP␣level␣remedy␣maidenkiss␣pheonixdown
```

In which:

- ***HP***: knight's health stat, which is an integer between 1 and 999. This value is also the knight's maximum health value ***MaxHP***.
- ***level***: knight's level, is an integer between 1 and 10.
- ***remedy***: The number of Remedy potions the knight carries, is an integer with a value between 0 and 99.
- ***maidenkiss***: The number of MaidenKiss potions the knight carries, is an integer with a value between 0 and 99.
- ***phoenixdown***: The number of PhoenixDown tears a knight has, is an integer with a value between 0 and 99.

The second line of the input file will describe the journey to Koopa. The trip to Koopa will include multiple events, each of which will be indexed starting at 1. Each event will be represented by a numeric value called an event code. The corresponding meaning of each event is described in Table 1. The number of events is not fixed and can change depending on the test case. An event can happen multiple times.

The third line of the input file will include the names of the files used in some events, the third line has the following format:

```
<file_mush_ghost>,<file_asclepius_pack>,<file_merlin_pack>
```

In which:

- <file_mush_ghost> is the file name used in the event 13<ms>.
- <file_asclipius_pack> is the file name used in the event 19.
- <file_merlin_pack> is the file name used in the event 18.

Details of each event are given in Section 4. 3 file names are separated by 1 comma. If the journey to Koopa does not have an event to use the corresponding file, the filename can be an empty string, the knight doesn't need to care about this file name.

Example 3.1

With input as:

```
172␣2␣0␣1␣1
1␣2␣9␣2␣18␣99
```

The Knight of the Round Table participating in the rescue of the princess will have ***HP*** of 172, level 2, no Remedy, 1 MaidenKiss, and 1 PhoenixDown. On the way to Koopa, the knight will encounter the following events in turn:

- Event 1: meet MadBear.
- Event 2: meet Bandit.
- Event 3: encounter another Bandit.
- Event 4: meet Merlin the wizard.
- Event 5: encounter the dragon Bowser.

Also, the knight's ***MaxHP*** is also 172 (meaning the knight's ***HP*** will never exceed 172 during the journey).

4 Description

During events processing, the functions must update the appropriate values for the 5 stats ***HP***, ***level***, ***remedy***, ***maidenkiss***, ***phoenixdown***. The ***rescue*** stat indicates whether the knight successfully rescued Princess Guinevere or not. ***rescue*** needs to be updated to the appropriate value after handling the events so that: ***rescue***= -1 if the journey is not over, ***rescue***= 1 if the journey is over and the rescue is successful, ***rescue***= 0 if the princess can't be rescued (e.g. knight's ***HP*** is less than 0 after fighting an opponent).

Table 1: Events on the journey to Koopa

Event code	Meaning
0	Bowser surrendered and handed over the princess
1	Meet MadBear
2	Meet Bandit
3	Meet LordLupin
4	Meet Elf
5	Meet Troll
6	Meet Shaman
7	Meet Siren Vajsh
11	Pick up the power-up mushroom MushMario
12	Pick up the Fibonacci mushroom MushFib
13<ms>	Pick up the ghost mushroom MushGhost
15	Obtain recovery potion Remedy
16	Obtain the antidote MaidenKiss
17	Pick up the phoenix tear PhoenixDown
18	Meet Merlin
19	Meet Asclepius
99	Meet Bowser

Example 4.1

With input as:

```
172␣2␣0␣1␣0
0
```

The Knights of the Round Table met with only one event: the dragon Bowser surrendered and returned the princess. The stats remain the same and have the following values:

- ***HP*** = 172
- ***level*** = 2
- ***remedy*** = 0
- ***maidenkiss*** = 1
- ***phoenixdown*** = 0
- ***rescue*** = 1

Depending on the events taking place along the way, the knight's ***HP***, ***level***, ***remedy***, ***maidenkiss***, and ***phoenixdown*** stats will be changed. Specifically, it is described as follows:

1. **If the event has code 0 is encountered**, the knight successfully rescues the princess,

and the knight's journey ends immediately, although there are still events else occur after this event in the input data.

Example 4.2

With input as

```
172␣2␣0␣1␣0
0␣10␣5␣15␣21␣99
```

The stat values will be:

- ***HP*** = 172
- ***level*** = 2
- ***remedy*** = 0
- ***maidenkiss*** = 1
- ***phoenixdown*** = 0
- ***rescue*** = 1

2. **If an event with codes 1 to 5 is encountered**, the knight must engage the corresponding opponent. Each opponent will also have their own ***levelO*** level. If an opponent is encountered at the **i** event (the first event corresponds to $i = 1$), the opponent's corresponding ***levelO*** will be calculated as follows:

$$b = i\%10 \quad (1)$$

$$levelO = i > 6?(b > 5?b : 5) : b \quad (2)$$

If the knight's ***level*** is higher than the opponent's ***levelO***, the knight wins. On victory, the knight's ***level*** will be increased by 1. However, if the knight's ***level*** is already 10 then ***level*** cannot be increased.

If the knight's ***level*** is equal to ***levelO***, the match is drawn, and the knight continues his journey but the ***level*** stats are not increased.

If the knight's level is less than the opponent's ***levelO***, the knight's ***HP*** will be recalculated as follows:

$$damage = baseDamage * \boldsymbol{levelO} * 10 \quad (3)$$

$$\boldsymbol{HP} = \boldsymbol{HP} - damage \tag{4}$$

Which, baseDamage will depend on the opponent, described in Table 2.
Note that $\boldsymbol{HP}$ will always be an integer when calculated using the formula 3 and 4. If $\boldsymbol{HP}$ is less than or equal to 0 after calculating with Equation 3 and the knight does not have the PhoenixDown phoenix tear, the knight will not be able to proceed to subsequent events, if any. Consequently, the program ends and the knight is not able to save the princess. In case the knight has PhoenixDown, this item will be automatically used (the $\boldsymbol{phoenixdown}$ will be reduced by 1), the knight's $\boldsymbol{HP}$ will be restored to the original $\boldsymbol{MaxHP}$ value and the knight continues his journey.

Table 2: baseDamage stats of opponents

Opponent	**baseDamage**
MadBear	1
Bandit	1.5
LordLupin	4.5
Elf	7.5
Troll	9.5

Example 4.3

With input as:

```
172␣1␣0␣1␣0
5␣2
```

In event 1, the knight encounters the troll with ***levelO*** of 1. Since the knight's ***level*** is currently 1, the match is drawn and the knight moves on to the next event. In event 2, the knight meets Bandit with ***levelO*** of 7. Since the knight's ***level*** is only 1, the knight lost, and the knight's ***HP*** will be left at $172 - 1.5 * 2 * 10 =$142. At this point, since there are no more events to follow, the knight reaches Koopa and successfully rescues the princess, the stats are:

- ***HP*** $= 142$
- ***level*** $= 1$
- ***remedy*** $= 0$
- ***maidenkiss*** $= 1$
- ***phoenixdown*** $= 0$
- ***rescue*** $= 1$

Example 4.4

With input as:

```
152␣1␣0␣1␣0
3␣5␣7
```

After a draw with LordLupin in event 1, the knight in event 2 fought Troll with ***levelO*** of 2 and lost due to having lower ***level***, the knight's ***HP*** will therefore become $152–8.5 * 2 * 10 = -18 < 0$. Thus, the program ends and the knight fails to rescue the princess. The final stats are:

- ***HP*** $= -18$
- ***level*** $= 1$
- ***remedy*** $= 0$
- ***maidenkiss*** $= 1$
- ***phoenixdown*** $= 0$
- ***rescue*** $= 0$

Example 4.5

With input as:

```
152␣1␣0␣1␣1
3␣5
```

Similar to the previous example, after a draw with LordLupin in event 1, in event 2, knights fought Troll with ***levelO*** of 2 and lost due to having lower ***level***, knight's ***HP*** due that would become $152–8.5 * 2 * 10 = -18 < 0$. Since the knight now has 1 PhoenixDown, this item will be used automatically, and the knight's ***HP*** restored to its original ***MaxHP*** value (152). At this point, since there were no further events, the knight reached Koopa and saved the princess. The final stats are:

- ***HP*** $= 152$
- ***level*** $= 1$
- ***remedy*** $= 0$
- ***maidenkiss*** $= 1$
- ***phoenixdown*** $= 0$
- ***rescue*** $= 1$

3. **If a Shaman is encountered (event code 6)**, the knight will engage the Shaman. Combat is the same as described in 2. If the knight wins, the knight's level will increase by 2 units, but the level cannot be more than 10. If it's a draw, the knight continues to advance. If lost, the knight will be turned into a tiny for the next 3 events. During the tiny state, the knight's ***HP*** will be reduced by 1/5 of the current ***HP*** (integers only). However, if the knight's initial ***HP*** is less than 5, the knight's ***HP*** will be counted as 1. After the tiny state expires, the knight's ***HP*** will again be multiplied by 5. After being multiplied, if ***HP*** now is greater than ***MaxHP***, ***HP*** will automatically decrease to ***MaxHP***.
   As soon as he is turned into a tiny, if the knight has the Remedy potion (remedy >= 1), the knight will automatically use this potion and return to the normal state, which ***HP*** is not changed, then the stat's remedy is reduced by 1. If the knight's ***HP*** drops to or below zero while the knight is in the tiny state, and the knight has PhoenixDown to use, the knight will be released from the tiny state, the knight's ***HP*** also will be restored to ***MaxHP***.

Example 4.6

With input as:

```
152␣1␣0␣0␣0
4␣6␣5
```

After a draw with Elf in event 1, the knight in event 2 fought Shaman with ***levelO*** of 2 and become tiny due to lower level, the knight's ***HP*** will then be reduced to 30. In event 3, the knight fought Troll with ***levelO*** of 3 and lost. The knight's ***HP*** is then reduced to $30 - 8.5 * 3 * 10 = -225 < 0$. Since the knight doesn't have PhoenixDown, the program ends and the knight fails to rescue the princess. The stats then will be:

- ***HP*** $= -225$
- ***level*** $= 1$
- ***remedy*** $= 0$
- ***maidenkiss*** $= 0$
- ***phoenixdown*** $= 0$
- ***rescue*** $= 0$

Example 4.7

With input as:

```
998␣1␣0␣0␣0
4␣6␣1␣1␣1
```

After a draw with Elf in event 1, the knight in event 2 fought Shaman with ***levelO*** of 2 and become tiny due to lower level, the knight's ***HP*** will then be reduced to 199. In events 3, 4, 5, the knight continuously fought with MadBears whose ***levelO*** was 3, 4, 5 respectively and all lost. The knight's ***HP*** is then reduced to 199-1*(3+4+5)*10 = 79. At this point, since 3 events have already passed, the knight's ***HP*** will automatically revert to 79*5= 395. Since there were no further events, the knight reached Koopa and successfully saved the princess. The final stats are:

- ***HP*** = 395
- ***level*** = 1
- ***remedy*** = 0
- ***maidenkiss*** = 0
- ***phoenixdown*** = 0
- ***rescue*** = 1

Example 4.8

With input as:

```
998␣1␣2␣0␣0
4␣6␣1
```

After a draw with Elf in event 1, the knight in event 2 fought Shaman with ***levelO*** of 2 and become tiny due to lower level, the knight's ***HP*** will then be reduced to 199. Since the knight has 2 ***remedy***, an antidote will be automatically used, then the knight's ***HP*** will be 199*5 = 995. In event 3, the knight defeated MadBear who has a ***levelO*** of 3 and the knight's ***HP*** after the fight is 995-1*3*10 = 965. The final stats are:

- ***HP*** = 965
- ***level*** = 1
- ***remedy*** = 1
- ***maidenkiss*** = 0
- ***phoenixdown*** = 0
- ***rescue*** = 1

4. **If encounters Siren Vajsh (event code 7)**, the knight engages Vajsh. Combat is the same as described in 2. If the knight wins, the knight's ***level*** will increase by 2 units (but cannot be more than 10), then will proceed to the next event. If it is a draw, the knight also continues to advance.
   If he loses, the knight will be turned into a frog in the next 3 events. As soon as he is turned into a frog, if the knight has MaidenKiss potion, the knight will automatically take this potion and return to normal, which the level is not changed and the ***maidenkiss*** stat will be reduced by 1. When turning into a frog, the knight's ***level*** will be reduced to 1. After transforming back into a human, the knight's *level* is restored to the previous value it was before being turned into a frog. Even if the knight's level is increased when being in the frog state, when turning into a human, the knight's level will also return to the previous value before being turned into a frog.
   While being transformed into a tiny or into a frog, if the knight picks up the corresponding antidote/potion (Remedy or MaidenKiss), the antidote/potion will be used automatically. Shaman and Vajsh will skip the fight with the knight if the knight is currently turned into a tiny or a frog.

Example 4.9

With input as:

```
998␣1␣0␣0␣0
4␣6␣7
```

After a draw with Elf in event 1, the knight in event 2 fought Shaman with ***levelO*** of 2 and become tiny due to lower level, the knight's ***HP*** will then be reduced to 199. In the third event, the knight meets Vajsh, due to the knight being turned into a minion, Vajsh will skip the fight with the knight. The stats then will be:

- ***HP*** $= 199$
- ***level*** $= 1$
- ***remedy*** $= 0$
- ***maidenkiss*** $= 0$
- ***phoenixdown*** $= 0$
- ***rescue*** $= 1$

5. **If the knight picks up the power-up mushroom MushMario (event code 11)**, the knight's ***HP*** will increase as follows:

$$\boldsymbol{HP} = \boldsymbol{HP} + (s1\%100)$$

In which: $n1 = ((\boldsymbol{level} + \boldsymbol{phoenixdown})\%5 + 1) * 3$, $s1$ is the sum of $n1$ the largest number in the set of 2-digit odd positive integers. For example, if $n1 = 3$ then $s1 = 99 + 97 + 95 = 291$.
Then ***HP*** is incremented to the nearest prime. After incrementing, if ***HP*** is greater than ***MaxHP***, ***HP*** will be automatically decremented to ***MaxHP***.

6. **If a knight picks up a Fibonacci mushroom MushFibo (event code 12)**, the knight's ***HP*** will drop to the nearest Fibonacci number if ***HP*** > 1. If ***HP*** $= 1$, this stat is not changed. A Fibonacci number is a number in the Fibonacci sequence. This is an infinite sequence of natural numbers starting with 1 and 1, then the next numbers will be the sum of the 2 preceding numbers.

7. **If a knight picks up a ghost mushroom MushGhost (event code starts with 13 followed by the string <ms>)**, the knight may encounter one or more ghost mushrooms varies based on the <ms> string. For example:

   - If the event code is 131, <ms> is 1, the knight encounters the Type 1 ghost mushroom.

- If the event code is 1342, <ms> is 42, the knight encounters Type 4 ghost mushrooms and then Type 2 ghost mushrooms. Immediately after encountering each type of ghost mushroom, the knight's ***HP*** needs to be updated for matching before meeting the next ghost mushroom. If ***HP*** exceeds ***MaxHP***, it also needs to be reinstalled ***HP*** with ***MaxHP*** before encountering the next ghost mushroom. If ***HP*** goes below 0; if potion ***phoenixdown*** > 0 then potion will be automatically used, ***HP*** is restored to ***MaxHP***, the knight will meet next ghost mushroom; if there are no more potions ***phoenixdown***, the program will stop, the knight does not meet the next ghost mushroom.

When encountering ghost mushrooms, it is necessary to manipulate a sequence of numbers stored in a file named <***file_mush_ghost***>. The file name is on line 3 of the input file (See item description 3). The file has a 2-line format: line 1 contains a positive integer $n2$, line 2 contains $n2$ integers separated by a comma, $n2 <= 100$. There are 4 types of ghost mushrooms corresponding to 4 codes 1, 2, 3, 4. The knight's ***HP*** will change for each type of mushroom as follows:

- ***Ghost mushroom type 1 (code 1)***: Let $maxi$ and $mini$ be the last position (position from 0) of the largest and smallest number, respectively in $n2$ integer in line 2.

$$\boldsymbol{HP} = \boldsymbol{HP} - (maxi + mini)$$

- ***Ghost mushroom type 2 (code 2)***: Checks if the number sequence has the shape of a mountain or not. A mountain-shaped number sequence is a sequence of numbers that increase to the largest number (call this number the peak) and then decrease. The ascending and descending property requires strict increment (a < b) and strict decrement (a > b) where a and b are 2 consecutive values in the array. The mountain shape accepts the top of the mountain on the first or last element. Let mtx and mti be the value and position of the mountain peak, respectively. If the sequence has no mountain shape, then $mtx = -2$, $mti = -3$.

$$\boldsymbol{HP} = \boldsymbol{HP} - (mtx + mti)$$

- ***Ghost mushroom type 3 (code 3)***: For each integer xi in the sequence in line 2 of the file, perform the transformation:
  - $xi = -xi$ if $xi < 0$

– $xi = (17 * xi + 9)\%257$

In the sequence of numbers after transformation: call $maxi2$ and $mini2$ to be the first position (from 0) of the largest number and the smallest number, respectively.

$$\boldsymbol{HP} = \boldsymbol{HP} - (maxi2 + mini2)$$

- ***Ghost mushroom type 4 (code 4)***: Perform number sequence transformation like Ghost mushroom type 3. Let $max2\_3x$ and $max2\_3i$ respectively be the second largest number and its first position (from 0) in the first 3 numbers in the sequence. If no second largest number exists then $max2\_3x = -5$, $max2\_3i = -7$.

$$\boldsymbol{HP} = \boldsymbol{HP} - (max2\_3x + max2\_3i)$$

8. **If the knight picks up Remedy, MaidenKiss or PhoenixDown (event codes 15, 16 and 17 respectively)**, the ***remedy***, ***maidenkiss*** and ***phoenixdown*** stats will be increased respectively to 1. However, these stats must not increase more than 99.
9. **If the knight encounters Asclepius (event code 19**, Asclepius is the god of potions, he will give the knight a bag of various potions such as ***remedy***, ***maidenkiss***, ***phoenixdown***. The information of the potion bag is stored in a file, the file name is stored in the variable ***file_asclepius_pack***, the file content format is as follows:

   - Line 1 contains a positive integer $r1$.
   - Line 2 contains an integer $c1$.
   - $r1$ next lines, each line contains $c1$ integers separated by 1 space. Each integer represents a potion. Asclepius only let the knight take the potion that is represented by an integer of 16, 17, 18 corresponding to ***remedy***, ***maidenkiss***, ***phoenixdown***. At the same time, Asclepius only allows the Knight to take up to 3 potions on a line, the way to get the potion is to go from the beginning to the end of each line.

   During the journey to Koopa, the knight can only be given the potion by Asclepius at most once and at the first encounter. If the knight encounters Asclepius the next time, the knight will ignore the god and do nothing. After receiving the potion, if the knight is in the tiny or frog state, and has the corresponding potion/antidote, the corresponding potion/antidote will be used automatically.
10. **If the knight's original *HP* was 999**, that knight was King Arthur who gave up the English throne to save Guinevere. Arthur will defeat all opponents in all skirmishes.
11. **If the knight's initial *HP* is a prime, the knight is a Lancelot.** Lancelot defeats all MadBear, Bandit, LordLupin, Elf, Troll, Shaman and Vajsh regardless of the opponent's

levelO.

12. **Knight may encounter Bowser himself during the journey (event code 99)**. Bowser can only be defeated by Arthur; or Lancelot of level 8 or greater; or the Knight of the Round Table whose level is 10. Upon defeating Bowser, the knight's level will be increased to 10. If he loses to Bowser, the program ends and the knight cannot rescue the princess. Note that the knight continues his journey even after defeating Bowser.
13. **If the knight encounters Merlin (event code 18)**, Merlin will give the knight an item bag. In this bag, there are a number of items enchanted by Merlin that can increase the knight's ***HP***. The information of the bag is stored in a file, the file name is stored in the variable ***file_merlin_pack***, the file content format is as follows:

    - Line 1 contains a positive integer $n9$.
    - $n9$ next lines, each line contains a string representing the name of the item. An enchanted item is an item whose name contains all 6 letters of Merlin's name, these 6 letters don't matter whether it's uppercase or lowercase.

    For each enchanted item, the knight gains 2 ***HP*** but cannot exceed ***MaxHP***. However, if the item name contains the string "Merlin" or "merlin", the knight is increased by 3 ***HP*** but not by more than ***MaxHP***.
    Merlin only showed the knight his bag when he first met him.

5 Requirements

To complete this assignment, students must:

1. Read entire this description file.
2. Download the initial.zip file and extract it. After extracting, students will receive files including main.cpp, main.h, knight.h, knight.cpp, and example file inputs. Students will only submit 2 files, knight.h and knight.cpp. Therefore, you are not allowed to modify the main.h file when testing the program.
3. Students use the following command to compile:
   g++ -o main main.cpp knight.cpp -I . -std=c++11
   Students use the following command to run the program:
   ./main tc1_input
   The above command is used in the command prompt/terminal to compile the program. If students use an IDE to run the program, students should pay attention: add all the files to the IDE's project/workspace; change the IDE's compile command accordingly. IDEs

usually provide buttons for compiling (Build) and running the program (Run). When you click Build, the IDE will run a corresponding compile statement, normally, only main.cpp should be compiled. Students need to find a way to configure the IDE to change the compilation command, namely: add the file knight.cpp, add the option -std=c++11, -I .

4. The program will be graded on the Unix platform. Students' backgrounds and compilers may differ from the actual grading place. The submission place on BKeL is set up to be the same as the actual grading place. Students must test the program on the submission site and must correct all the errors that occur at the BKeL submission site in order to get the correct results when final grading.
5. Modify the files knight.h, knight.cpp to complete this assignment and ensure the following two requirements:
   - Implement function **adventureToKoopa** describing the knight's journey to Koopa and encountering various events. Event handling method as described in the specifications. In this function, the parameter **file_input** contains the name of the input file for the program, the rest of the parameters have the same meaning as the knight stats and the ***rescue*** stat. In the **adventureToKoopa** function, after each event, the program needs to print out the current stats by calling the provided `display` function.
   - There is only one **include** directive in the knight.h file, which is **#include "main.h"**, and one directive in the knight.cpp file, which is **#include "knight.h"**. Apart from the above directives, no other **#include** is allowed in these files.
6. Students are encouraged to write additional functions to complete this assignment.

6 Submission

Students submit only 2 files: knight.h and knight.cpp, before the deadline given in the link "Assignment 1 - Submission". There are a number of simple test cases used to check student work to ensure that student results are compilable and runnable. Students can submit as many times as they want, but only the final submission will be graded. Since the system cannot bear the load when too many students' submissions at once, students should submit their work as soon as possible. Students do so at their own risk if they submit assignments by the deadline. When the submission deadline is over, the system will close so students will not be able to submit any more. Submissions through other means will not be accepted.

7 Other regulations

- Students must complete this assignment on their own and must prevent others from stealing their results. Otherwise, the student treat as cheating according to the regulations of the school for cheating.
- Any decision made by the teachers in charge of this assignment is the final decision.
- Students are not provided with test cases after grading, students will be provided with the assignment's score distribution.
- Assignment contents will be harmonized with a question in exam with similar content.

END