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

SHERLOCK
A STUDY IN PINK - Part 1

Version 1.3

Ho Chi Minh City, 02/2023

ASSIGNMENT'S SPECIFICATION

Version 1.1

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

2 Introduction

This assignment is based on episode 1 season 1 of a TV series on BBC named Sherlock. This series is a British mystery crime drama television series based on Sir Arthur Conan Doyle's Sherlock Holmes detective stories.

John Watson is a military medic resting in London after being wounded in Afghanistan. He was thinking about changing to another apartment with a lower rent when he accidentally met an old friend. Watson was then introduced to Sherlock Holmes by that friend to share a room at 221B Baker Street, owned by Mrs. Hudson. From there, Watson was drawn into Sherlock's challenging cases, and Sherlock had a new companion in his quest to solve the case.

The most recent case was of a woman in a pink dress. This case is different from previous cases in that: the victim scratched the floor with her fingernails and left a message. The results of the scene investigation showed that the victim lost her luggage. With Sherlock's talent, he found the lost luggage. Thereby, he took it to room 221B Baker Street and looked for further traces of the criminal. Watson also returns at the same time and joins Sherlock.

3 Input data

The input data is contained in a file, the file name will be passed into the program through the variable **file_input**. This file will contain the following information:

```
HP1␣HP2
EXP1␣EXP2
M1␣M2
E1␣E2␣E3
```

In detail:

- $\mathbf{HP_1}$ and $\mathbf{HP_2}$ is the health points of Sherlock and Watson, respectively, is an integer in [0, 666]. In any calculation case, if the HP is over 666, you must set it to 666. Otherwise, if HP is less than 0, it must be set to 0.
- $\mathbf{EXP_1}$ and $\mathbf{EXP_2}$ is the experience points when solving the cases of Sherlock and Watson, respectively, is an integer in [0, 600]. The more clues is found, the higher the experience points are. In any calculation case, if the HP is over 600, you must set it to 600. Otherwise, if HP is less than 0, it must be set to 0.
- $\mathbf{M_1}$ and $\mathbf{M_2}$ is the initial money of Sherlock and Watson, respectively, is an integer in [0, 3000]. If the money is over 3000, you must set it to 3000. Otherwise, if the money is less than 0, it must be set to 0.
- **E** are the event codes of each missions in this assignment, respectively, is an integer in [0, 99].

  Note:

  - In any case if the calculation results in a non-integer for **HP**, **EXP** and **M**, that number must be rounded up immediately.
  - Each mission will have its case, if $\mathbf{E_i}$ is **outside the range** given in the case of the mission, the mission function won't do anything (so don't change input parameter), and will return -99.

4 Missions

Students are asked to build a program in C++ to simulate the first case of Sherlock and Watson: A study in Pink, through the tasks described below.

4.1 Mission 1: The first encounter (1.5 points)

In the first meeting, Watson witnessed Sherlock's genius deductive ability. Sherlock guesses Watson's return from Afghanistan and other details behind Watson's life. Students were asked to write a function to describe the process that Sherlock explains to Watson his deductions. Through this process, Watson's and Sherlock's **EXP** will change:

- Function name: firstMeet.
- Input parameters:
  - $\mathbf{EXP_1}$: Sherlock's experience points.
  - $\mathbf{EXP_2}$: Watson's experience points.
  - $\mathbf{E_1}$: Code that identify event 1
- Return value: The integer is the sum of $\mathbf{EXP_1}$ and $\mathbf{EXP_2}$

Note: In the function in this and later mission, the parameters representing the mutable indices will be passed by reference. When there is a request to update the variables, students need to make updates on these reference variables. Then, the variables passed in will also be updated accordingly.

4.1.1 Case 1 (0.5 points)

In the case of $\mathbf{E_1}$ in range [0, 3], Sherlock explains how he knows Watson has just returned from Afghanistan. Here are the things Sherlock describes that change Watson's **EXP**:

Information	$\mathbf{E_1}$'s value	Sherlock's Observation	Outcome
1	0	Watson's hairstyle and manner of speech are like those of the military	Add 29 EXP
2	1	His face is tanned but not tanned under the wrist, proving that Watson has returned from abroad	Add 45 EXP
3	2	Watson limped, but when they met, he chose to stand without asking for a chair, so he had psychological problems after being injured. This could be an injury caused by action on the battlefield	Add 75 EXP
4	3	Sherlock explained all the 3 information	Add EXP that equal to a total of Information 1 and Information 2 and Information 3

From the above information, Sherlock guessed that Watson was a military doctor returning from abroad, his search range was reduced to 2 countries: Afghanistan or Iraq. In this case, Sherlock's decision (D integer) will be depend on factors including event code $\mathbf{E_1}$ and Sherlock's experience $\mathbf{EXP_1}$ following the relation: $\boldsymbol{D}$=$\boldsymbol{E_1} * 3 + \boldsymbol{EXP_1} * 7$

If D is an even number, Sherlock will make a prediction in favor of Afghanistan (and it is a true decision), then his **EXP** will be added by an amount of $D/200$. Conversely, if D is an odd number, Sherlock will lean towards the possibility of Iraq, which is a bad choice and his EXP is reduced by $D/100$.

Example 4.1

With the input data:

```
172␣172
400␣300
450␣450
3␣1␣0
```

With the event code $\mathbf{E_1} = 3$, that means Sherlock will explain based on all the 3 information

So that: $\mathbf{EXP_2} = 300 + 29 + 45 + 75 = 449$

Decision value: $D = 3 * 3 + 400 * 7 = 2809$ is an odd number

So that $\mathbf{EXP_1}$ will be reduced to: $\mathbf{EXP_1} = 400 - (2809/100) = 371.91 \xrightarrow{\text{Round upto}} 372$

The return value of this function: $output = \mathbf{EXP_1} + \mathbf{EXP_2} = 449 + 372 = 821$

4.1.2 Case 2 (1.0 điểm)

In case that $\mathbf{E_1}$ is in the range of $[4, 99]$, Sherlock will explain why he has known that Watson got a brother. Below is the information that Sherlock gave making Watson's **EXP** change:

Information	$\mathbf{E_1}$'s range	Sherlock's Observation	Outcome
1	[4, 19]	Watson has an expensive phone but he is looking for a roommate to share the rent, the phone must have been given to Watson by someone else	Add ($\mathbf{E_1}$/4+19) EXP.
2	[20, 49]	The phone has many scratches indicating that it has been placed with many other items such as keys, coins. Watson wouldn't do that to a luxury item. This is caused by the previous owner with the phone	Add ($\mathbf{E_1}$/9+21) EXP.
3	[50, 65]	On the phone is engraved the name: Harry Watson, showing that this was given to him by an old family member	Add ($\mathbf{E_1}$/16 + 17) EXP.
4	[66, 79]	Sherlock explains information 1; after Watson finished listening and $\mathbf{EXP_2}$ was updated, if $\mathbf{EXP_2} > 200$, Sherlock continued to interpret the information 2 and Watson was updated with $\mathbf{EXP_2}$ respectively.	(As described).
5	[80, 99]	Sherlock explains information 1 and 2; after Watson finished listening and $\mathbf{EXP_2}$ was updated, if $\mathbf{EXP_2} > 400$, Sherlock continued to interpret the information 3 and Watson was updated with $\mathbf{EXP_2}$ respectively.	(As described).

Note: If Watson is explained by Sherlock with all 3 information 1, 2 and 3, Watson will be added 15% of the current EXP (after updating the EXP for all 3 information)

After Sherlock explained the information to Watson, Watson said: "Harry stands for Harriet". Thus, Harry is Watson's sister, not his brother. Surprised by this mistake, Sherlock's EXP is reduced by an amount of $\mathbf{E_1}$ (EXP).

Example 4.2

With $\mathbf{EXP_1} = 500$, $\mathbf{EXP_2} = 450$, $\mathbf{E_1} = 40$. Following the second information, we have:

$$\mathbf{EXP_2} = \mathbf{EXP_2} + (\mathbf{E_1}/9 + 21) \approx 475.44 \xrightarrow{\text{Round upto}} 476$$

$\mathbf{EXP_1}$ will be reduced to:

$$\mathbf{EXP_1} = \mathbf{EXP_1} - \mathbf{E_1} = 500 - 40 = 460$$

Output of the function:

$$\text{output} = 476 + 460 = 936$$

Example 4.3

With $\mathbf{EXP_1} = 500$, $\mathbf{EXP_2} = 450$, $\mathbf{E_1} = 81$. Following the fifth information, $\mathbf{EXP_2}$ first would be updated according to the first and second information, we have:

$$\mathbf{EXP_2} = \mathbf{EXP_2} + (\mathbf{E_1}/4 + 19) \approx 489.25 \xrightarrow{\text{Round upto}} 490$$

$$\mathbf{EXP_2} = \mathbf{EXP_2} + (\mathbf{E_1}/9 + 21) = 520$$

As $\mathbf{EXP_2} > 400$ so Sherlock has continuously explained the third information and the $\mathbf{EXP_2}$ would be increased to:

$$\mathbf{EXP_2} = \mathbf{EXP_2} + (\mathbf{E_1}/16 + 17) = 542.06 \xrightarrow{\text{Round upto}} 543$$

As Watson has been explained all the three information, $\mathbf{EXP_2}$ would be risen 15%:

$$\mathbf{EXP_2} = \mathbf{EXP_2} * 1.15 \xrightarrow{\text{Round upto}} 624.45 \xrightarrow{\text{Greater than 600}} 600$$

$\mathbf{EXP_1}$ would be decreased to:

$$\mathbf{EXP_1} = \mathbf{EXP_1} - \mathbf{E_1} = 419$$

The output of this function:

$$\text{output} = 600 + 419 = 1019$$

4.2 Mission 2: Tracing the luggage (2.5 points)

After their first meeting, Watson was surprised at Sherlock's genius deductive abilities. The very next day, the two went to see Mrs. Hudson's apartment at 221B Baker Street. At that time, Lestrade - the inspector in charge of these suicide cases came and asked Sherlock for helping him to follow up the case. Through the investigation of the crime scene, Sherlock discovered that the victim had traveled from a place where it was raining to here and had lost their luggage. After checking the weather, Sherlock found out that the nearest place where it was raining was Cardiff. He began to try to find the routes from Cardiff to the crime scene to search for the victim's luggage.

Students are asked to write a function to describe Sherlock's luggage searching process, the function information is described as follows:

- Name: traceLuggage.
- Input parameter:
  - Sherlock's health points: $\mathbf{HP_1}$
  - Sherlock's experience points: $\mathbf{EXP_1}$
  - Sherlock's money: $\mathbf{M_1}$
  - Event code: $\mathbf{E_2}$
- Return value: $\mathbf{HP_1} + \mathbf{EXP_1} + \mathbf{M_1}$

After ruling out the possibilities, Sherlock found 3 possible roads the criminal took the victim and it is possible that he will dump the luggage on the roadside when he discovers the luggage in the car. Sherlock must try each route to find the lost luggage.

For each road, it depends on **EXP** and **HP** of Sherlock that would make difference probabilities. Details:

4.2.1 Road 01

Let S be the nearest perfect square to the value $\mathbf{EXP_1}$. If $\mathbf{EXP_1} \geq S$, the probability for Sherlock to find the suitcase on this route is:

$$P_1 = 100\%$$

Otherwise:

$$P_1 = \left(\frac{\mathbf{EXP_1}}{S} + 80\right)/123$$

(Note: P value does not need to be rounded up in calculation)

4.2.2 Road 02

On this route, Sherlock needs to spend money at various events along the way. With his amount of money $\mathbf{M_1}$, Sherlock needs to go through the following events:

- If $\mathbf{HP_1} < 200$, Sherlock will get into a grocery store to buy some food and beverage to recover his health. At that time Sherlock's **HP** will be added an amount equal to 30% of the existing **HP**, and his budget will be reduced by 150. If $\mathbf{HP_1}$ is not smaller than 200, Sherlock just needs some water and at that time **HP** will be added an amount equal to 10% of the existing **HP**, and his budget will be reduced by 70.
- The distance needed to travel was quite long, so Sherlock needs to rent a taxi or carriage. The price to rent a taxi to cover this distance would be 200, and a horse-drawn carriage would cost 120. If Sherlock's **EXP** is < 400, Sherlock will choose to take a taxi, otherwise he will take a horse-drawn carriage. At this time, Sherlock's **EXP** would be increased by 13%.
- After that, Sherlock met a homeless person and this person promised to reveal the clue where he saw the suitcase to Sherlock if Sherlock helped him with some money. If Sherlock's **EXP** < 300, Sherlock will believe and help this homeless person with $m = 100$ and listen to this person's instructions. If Sherlock's **EXP** is 300 or more, Sherlock will help with $m = 120$ and ask this person to lead the way. Even so, the homeless person mistook it for another empty suitcase. Sherlock's **EXP** will be reduced by 10%.

If $\mathbf{E_2}$ is an odd number, these events would be continuously repeated until at a completed event point, the number of paid money is greater than 50% the total at the time he started this second route, after that, he would just walk till the end of this road and not meet any events else. At that time, $\mathbf{HP_1}$ would be reduced by 17% and $\mathbf{EXP_1}$ would be raised by 17%. In contrast, if $\mathbf{E_2}$ is an even number, Sherlock would just do one round of those actions and keep walking to the end. In this case, if M is not enough for this one round of those actions, Sherlock will stop after finishing the event that makes the Sherlock's budget be 0. **EXP** and **HP** will still update as if $\mathbf{E_2}$ is odd.

The probability P_2 of finding the suitcase on this route will be calculated at the end of the route and calculated using the formula mentioned in route 01.

4.2.3 Road 03

Given a fixed array of numbers consisting of 10 elements, there are 10 probability values

$$P = \{32, 47, 28, 79, 100, 50, 22, 83, 64, 11\}$$

Let i be the index value of the probability P_i that Sherlock finds the suitcase on this road (i is indexed from 0). If $\mathbf{E_2}$ is a one-digit number, that value is the value of i. If $\mathbf{E_2}$ is a 2-digit number, calculate the sum of those 2 digits and take the number of the unit place of this total value as the value for i.

After going through all 3 routes, if all 3 routes that Sherlock has gone through have a probability of 100%, it means that Sherlock has made a mistake somewhere and needs to recalculate. At this time $\mathbf{EXP_1}$ is reduced by 25%. If not all are 100%, the average of the 3 probability values is the final probability of finding the suitcase. If this value is less than 50%, Sherlock will have a hard time finding the suitcase, so in the end, $\mathbf{HP_1}$ will decrease by 15% and $\mathbf{EXP_1}$ will increase by 15%. Conversely, if this value is greater than or equal to 50%, Sherlock will quickly find the suitcase, so in the end, $\mathbf{HP_1}$ will decrease by 10% and $\mathbf{EXP_1}$ will increase by 20%. (**Note: EXP** and **HP** are calculated on the value after going through all 03 routes).

Example 4.4

With $\mathbf{E_2} = 39$, $\mathbf{HP_1} = 333$, $\mathbf{EXP_1} = 430$, $\mathbf{M_1} = 890$

We have: The nearest perfect square to $\mathbf{EXP_1}$ is $S = 441$, so:

$P_1 = (\frac{430}{441} + 80)/123 = 0.66$

On the road 02: 50% of the initial $\mathbf{M_1}$: $890 * 0.5 = 445$

$\mathbf{E_2} = 39$ is an odd number, so that:

- Since $\mathbf{HP_1} = 333 > 200$ so: $\mathbf{M_1} = 890 - 70 = 820$
  $\mathbf{HP_1} = 333 * 1.1 = 366.3 \xrightarrow{\text{Round upto}} 367$
  Total paid money: $70 < 445$
- As $\mathbf{EXP_1} = 430 > 400$ so: $\mathbf{M_1} = 820 - 120 = 700$
  $\mathbf{EXP_1} = 430 * 1.13 = 485.9 \xrightarrow{\text{Round upto}} 486$
  Total paid money: $70 + 120 = 190 < 445$
- Because $\mathbf{EXP_1} = 486 > 300$ so: $\mathbf{M_1} = 700 - 120 = 580$
  $\mathbf{EXP_1} = 486 * 0.9 = 437.4 \xrightarrow{\text{Round upto}} 438$
  Total paid money: $190 + 120 = 310 < 445$

Keep repeating the events:

- Since $\mathbf{HP_1} = 367 > 200$ so: $\mathbf{M_1} = 580 - 70 = 510$
  $\mathbf{HP_1} = 367 * 1.1 = 403.7 \xrightarrow{\text{Round upto}} 404$
  Total paid money: $310 + 70 = 380 < 445$
- As $\mathbf{EXP_1} = 540 > 400$ so: $\mathbf{M_1} = 510 - 120 = 390$
  $\mathbf{EXP_1} = 438 * 1.13 = 494.94 \xrightarrow{\text{Round upto}} 495$
  Total paid money: $380 + 120 = 500 > 445$
  At this point Sherlock will stop and just walk to the end of the street

$\mathbf{HP_1} = 404 * 0.83 = 335.32 \xrightarrow{\text{Round upto}} 336$

$\mathbf{EXP_1} = 495 * 1.17 = 579.15 \xrightarrow{\text{Round upto}} 580$

The nearest perfect square to $\mathbf{EXP_1}$ is 576 so that $P_2 = 100\%$

On the road 03: With $\mathbf{E_3} = 39$

Sum of the two digits: 3+ 9 = 12 Therefore: i = 2

Thereby $P_3 = P[2] = 28\%$

Average probability: $P = \frac{P_1+P_2+P_3}{3} = 65\% > 50\%$

So that:

$\mathbf{HP_1} = 336 * 0.9 = 302.4 \xrightarrow{\text{Round upto}} 303$

$\mathbf{EXP_1} = 580 * 1.2 = 696 \xrightarrow{\text{Greater than 600}} 600$

Output result: $303 + 600 + 390 = 1293$

4.3 Mission 3: Chase the taxi (3 points)

After having found the luggage, Sherlock thought that: The victim will carry his cell phone. The phone was not at the scene, and neither was it in the luggage. So, it's most likely with the criminal. Sherlock tells Watson to send a text message to the victim's phone, and tells him that she just woke up from fainting and didn't know what happened. After that, they made an appointment with the person holding the phone to meet at an address.

After making an appointment to meet the person holding the phone, Sherlock is confident that he would be worried to hear that the victim was still alive if it were a criminal. The criminal will come to the arrangement point to see the actual condition of the victim. Sherlock and Watson went to a roadside shop about 5m away from the meeting point and watched together. A taxi came and stopped there, the person sitting in the taxi looked out searching. When this person accidentally looks in Sherlock's direction, the car starts up and leaves. Sherlock knows the streets of the city he lives in well. He and Watson ran through the shortcuts and chased the taxi.

Students are asked to write the following function to describe this process. The function information is as follows:

- Function name: **chaseTaxi**
- Input parameters:
  - Details about Sherlock và Watson, repsectively: $\mathbf{HP_1}, \mathbf{EXP_1}$ and $\mathbf{HP_2}, \mathbf{EXP_2}$
  - Event code $\mathbf{E_3}$
- Function requirements:
  - Initialize a $10matrix$x10 with each element of the array being an integer initialized to the value 0. This 2-dimensional array represents the map that that taxi chased with Sherlock and Watson. according to. The value of each location is the skill score of that taxi at that location.
  - Starting at position (0,0), the taxi moves in a row-by-row direction. For each time passing a point, the taxi's score at that point is equal to $((\mathbf{E_3} * j) + (i * 2)) * (i - j)$ (where i is the row index and j is the column index)
  - We define the left diagonal of a matrix at position X(i, j) as the path along the diagonal direction from a position in the first row or first column such that it passes through X and ends at a position in the last row or last column. The diagonal direction is defined as the direction in which both row and column indices change sequentially by 1 unit (either increasing or decreasing). Conversely, the right diagonal

of a matrix at position X(i, j) is the path along the diagonal direction from a position in the last row or last column such that it passes through X and ends at a position in the first row or first column.

For example, we have a matrix:

1	2	3
4	5	6
7	8	9

The left diagonal of the point with coordinates (1,0) includes positions (1,0) and (2,1), meaning the values are 4 and 8 respectively. Meanwhile, the right diagonal of the point with coordinates (1,2) includes positions (1,2) and (2,1), meaning the values are 6 and 8 respectively. (Note: Row and column indices start from 0, rows go from top to bottom, columns go from left to right.)

- Meanwhile, Sherlock and Watson take shortcuts (moving along each column). We also need a matrix to store their points. Their score at each location is equal to the **maximum value** from both the left diagonal and right diagonal of the taxi that passed through that point (all values for the taxi need to be calculated at each position before computing the scores for Sherlock and Watson). If this score is negative, take its absolute value.
- Taxi would meet Sherlock and Watson at a point (i, j). With i is equal to the number of points that have the value is greater than $\mathbf{E_3} * 2$. And j is equal to the number of points that have the value is smaller than -$\mathbf{E_3}$. If i or j is a two-digits number, keep repeating the stage sum of two digits until the sum is the one-digit number.
- At this meeting location, if **the absolute value of** the taxi's point is greater than Sherlock and Watson's, they would not catched the taxi. Otherwise, the would catch the taxi.
- If they can catch the taxi, **EXP** and **HP** of each guys will increase by 12% and 10%. Otherwise, they will decrease by 12% and 10%.

- Return result: Function returns the greater grade between taxi and Sherlock & Watson (Note that: Return the negative value if the grade is negative).

Example 4.5

With $\mathbf{E_3} = 59, \mathbf{HP_1} = 400, \mathbf{EXP_1} = 600, \mathbf{HP_2} = 350, \mathbf{EXP_2} = 500$

The grade matrix of the taxi is:

0	-59	-236	-531	-944	-1475	-2124	**-2891**	-3776	**-4779**
2	0	-120	-358	-714	-1188	-1780	-2490	**-3318**	-4264
8	63	0	-181	-480	-897	-1432	**-2085**	-2856	**-3745**
18	130	124	0	-242	-602	**-1080**	-1676	-2390	-3222
32	201	252	185	0	**-303**	-724	-1263	-1920	-2695
50	276	384	374	**246**	0	-364	-846	-1446	-2164
72	355	520	**567**	496	307	0	-425	-968	-1629
98	438	**660**	764	750	618	368	0	-486	-1090
128	**525**	804	965	1008	933	740	429	0	-547
162	616	952	1170	1270	1252	1116	862	490	0

The number of positive values in the matrix greater than $(\mathbf{E_3} * 2)$ is 37

So that: $i = 3 + 7 \rightarrow 10 \rightarrow 1 + 0 = 1$

The number of negative values in the matrix smaller than $(-\mathbf{E_3})$ is 44

So that: $j = 4 + 4 = 8$

Meeting location is at: (1, 8)

At this point, Taxi's grade is -3318

At this point, Sherlock & Watson's grade is 660

As abs(-3318) > 660 so Sherlock and Watson cannot catch the taxi. The values are updated:

$$\mathbf{EXP_1} = \mathbf{EXP_1} * 0.88 = 528$$

$$\mathbf{HP_1} = \mathbf{HP_1} * 0.9 = 360$$

$$\mathbf{EXP_2} = \mathbf{EXP_2} * 0.88 = 440$$

$$\mathbf{HP_2} = \mathbf{HP_2} * 0.9 = 315$$

Output value: -3318

Example 4.6

With $\mathbf{E_3} = 99, \mathbf{HP_1} = 400, \mathbf{EXP_1} = 600, \mathbf{HP_2} = 350, \mathbf{EXP_2} = 500$

After calculating the grade matrix at each point of the taxi, we have: The number of positive values in the matrix greater than $(\mathbf{E_3} * 2)$ is 35

So: $i = 3 + 5 = 8$

The number of negative values in the matrix smaller than $(-\mathbf{E_3})$ is 44

Therefore: $j = 4 + 4 = 8$

Meeting location is at: (8, 8)

At this point, Taxi's grade is 0

At this point, Sherlock & Watson's grade is 1442

As abs(0) $<$ 1442 so that Sherlock and Watson will catch the taxi successfully. The values are updated:

$$\mathbf{EXP_1} = \mathbf{EXP_1} * 1.12 = 672$$

$$\mathbf{HP_1} = \mathbf{HP_1} * 1.1 = 440$$

$$\mathbf{EXP_2} = \mathbf{EXP_2} * 1.12 = 560$$

$$\mathbf{HP_2} = \mathbf{HP_2} * 1.1 = 385$$

Output value: 1442

4.4 Mission 4: Valid Password (1.5 points)

After chasing the taxi, Sherlock and Watson returned to the apartment to rest. Sherlock noticed the laptop found in the luggage. Outside the luggage, there was a card with the victim's email address. He tried using this email as the login name, and for the password, he tried all the usual passwords, but all failed. In the luggage, Sherlock also found a notebook recording many different character strings, which could be the password to log in. However, most of these violate the rules for setting passwords for laptops.

As many password strings could be, students are asked to write a function to check the validity for each string. The function information is as follow:

- Function name: **checkPassword**.
- Input parameters:
  - `s`: string that needs to check if it is valid or not
  - `email`: email string of the victim

- Function requirements: Let *se* be the string of characters before the '@' character in the victim's email. This email will be guaranteed to have exactly 1 '@' character and has a maximum length of 100 characters. A valid password requirement is that it must simultaneously satisfy the following conditions:
  - has a minimum of 8 characters and a maximum of 20 characters.
  - Each character in the password can only be a number, or a lowercase letter, or an uppercase letter, or a special character. A special character is one of the following characters: '@', '#', '%', '$', '!'.
  - Must not contain the string *se*.
  - Must not contain more than 2 consecutive and identical characters.
  - Must contain at least 1 special character.
- Return results:
  - If **s** is a valid password, return -10.
  - If **s** has a length shorter than the minimum length, return -1.
  - If **s** has a length longer than the maximum length, return -2.
  - If **s** contains **se**, then return -(300+<sei>), where <sei> is the first occurrence of **se**.
  - If **s** contains more than 2 consecutive and identical characters, then return -(400+<sci>), where <sci> is the first position of the first string containing more than 2 consecutive and identical characters.
  - If **s** does not contain special characters, return -5.
  - The remaining cases return the position of the first character in violation of the validity requirement stated in *Function Requirements*.
- Note: if multiple conditions are violated, the function will return the condition that appears first in the *Return results* section.

Example 4.7

With `email = "pink@gmail.com", s = "123xyz"`.

We have: `se = "pink"`. The string s has the length of 6, shorter than the minimum length 8

Return result: -1.

Example 4.8

With `email = "pink@gmail.com"`, `s = "012345pink#pink"`.

We have: `se = "pink"`. The string s that contains se in the first occurrence is <sei> = 6.

Return result: $-(300+6) = -306$.

4.5 Mission 5: Find the correct password (1.5 points)

After eliminating invalid passwords, there are still quite a few passwords that need to be tried to find the correct password for the Laptop. Sherlock noticed that there were similar passwords appearing. Sherlock guessed that the password that appeared the most times was the Laptop password.

Students are asked to write a function to find the correct laptop's password. The function information is as below:

- Function name: **findCorrectPassword**.
- Input parameters:
  - `arr_pwds`: an array of passwords
  - `num_pwds`: the number of passwords in the array `arr_pwds`.
- Return result: The first position of the correct password in the array `arr_pwds`. A correct password is the password that appears in the `arr_pwds` array the most times and has the longest length among the most frequently occurring passwords. Since there can be multiple correct passwords, the function returns the position of the first occurrence of any correct password in the array.
- Note: Testcases are guaranteed to have up to **30** distinct passwords in the array `arr_pwds`.

Example 4.9

With
`arr_pwds = {"1234#xyz", "pink#pink", "pink@123", "123!pink", "pink#pink", "pink@123", "pink@123"}`.
We have:

- Password `"pink@123"` appears 3 times.
- Passowrd `"pink#pink"` appears 2 times.
- Password `"1234#xyz"` appears 1 time.
- Password `"123!pink"` appears 1 time.

Therefore, the password `"pink@123"` has the highest number of occurrences. The first position of `"pink@123"` in `arr_pwds` is 2.
Return result: 2.

Example 4.10

With
`arr_pwds = {"pink@123", "123!pink", "1234#xyz", "pink#pink", "pink#pink", "pink@123"}`.
We have:

- Password `"pink@123"` appears twice and has a length of 8.
- Password `"pink#pink"` appears twice and has a length of 9.
- Password `"1234#xyz"` appears once.
- Password `"123!pink"` appears once.

The two passwords `"pink@123"` and `"pink#pink"` have the highest number of occurrences (both equal to 2). But the password `"pink#pink"` is longer than `"pink@123"` so the correct password is `"pink#pink"`. The first position of `"pink#pink"` in `arr_pwds` is 3.
Return result: 3.

4.6 Ending

After successfully booting up the laptop, on the victim's desktop, there were only a few basic office software and a tracking software. This software was installed to connect to the victim's phone. Fortunately, both the username and password of the software matched the information used to log in to the laptop. The software began its search, the area on the screen started to zoom in on the scope and displayed the address 221B Baker Street. Mrs. Hudson rushed to flag down a taxi parked outside, the driver asked her to relay a message: 'Taxi reserved for Sherlock Holmes.' Sherlock suddenly understood everything, he told Watson to stay behind and continue searching, he needed to go out and get some fresh air.

What will Sherlock do to deal with the taxi driver outside, who is likely the culprit behind the 4 suicide cases? Will Watson catch up with Sherlock and assist him in defeating this criminal? We will continue to carry out new tasks with Sherlock and Watson in Part 2 of this Assignment.

Have fun doing assignment!!!

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, study_in_pink1.h, study_in_pink1.cpp, and example file inputs. Students will only submit 2 files, study_in_pink1.h and study_in_pink1.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 study_in_pink1.cpp -I . -std=c++11
   Students use the following command to run the program:
   ./main sa_tc_01_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 LMS 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 LMS submission site in order to get the correct results when final grading.

5. Modify the files study_in_pink1.h, study_in_pink1.cpp to complete this assignment and ensure the following two requirements:
   - There is only one **include** directive in the study_in_pink1.h file which is **#include "main.h"** and one directive in the knight.cpp file which is **#study_in_pink1** . Apart from the above directives, no other **#include** is allowed in these files.
   - Implement the functions described in the Missions in this Assignment.
6. Students are encouraged to write additional functions to complete this assignment.

6 Submission

Students submit only 2 files: study_in_pink1.h và study_in_pink1.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.

8 Cheating treatment

Assignment must be done by students themselves. Students will be considered cheating if:

- There is an unusual similarity between the source code of the submissions. In this case, ALL submissions will be considered fraudulent. Therefore, students must protect their source code.
- Student's work is submitted by another student.
- Students do not understand the source code they write, except for the code parts provided in the initialization program. Students can refer to any source, but must ensure that they clearly understand the meaning of all the commands they write. In case they do not clearly understand the source code from the place they refer to, students are specifically warned NOT to use this source code; Instead, they should use what they have learned to write programs.
- Wrongly submitted another student's work to the other personal account.
- Using source code from tools capable of generating source code without understanding the meaning.

In case of being concluded as cheating, the student will receive a score of 0 for the entire subject (not just this assgingment).

DO NOT ACCEPT ANY INTERPRETATION AND NO EXCEPTIONS!

After each assignment is submitted, a number of students will be randomly called for an interview to prove that the assignment just submitted was done by themselves.

9 Changelog

- Edit the example of the right diagonal of the matrix in section 4.3
- Add password conditions in section 4.4
- Add note about rounding up P values
- Adjust example 4.6.
- Adjust the input passwords in task 5 to be correct passwords.