Assignment 0

KING ARTHUR AND THE KNIGHTS OF THE ROUND TABLE

1. Introduction

King Arthur is a legendary king in the English history, who led the Knights of the Round Table to fight for justice. Almost invincible on the battlefield, King Arthur and the Knights of the Round Table only met worthy opponents when confronting with the Saxon warriors, led by King Cerdic, in the battle to decide the throne of the country. Due to the chivalry code, they decided to organize duels between representatives of the two sides to resolve all disputes.

The Knights of the Round Table and the Saxon warriors are all good fighters, each duel between them is always a fierce battle, but the victory or defeat is sometimes not only determined by the talents of the fighter, but also by other factors such as weapons, armor, battlefield, etc. Normally, such a battle would only end when one of the two fighters fatally wounded.

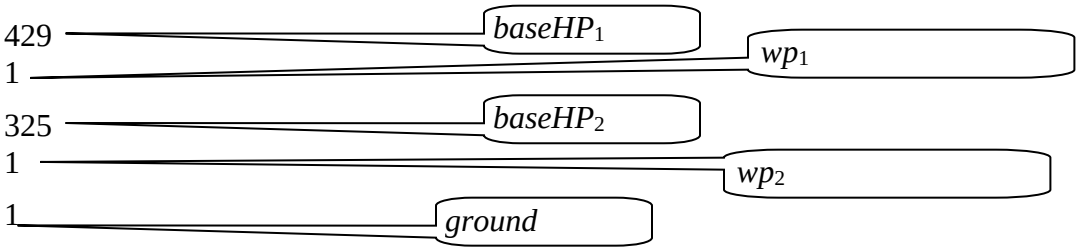
Therefore, before each hand-to-hand battle, King Arthur always worried about the fate of his brother's Round Table knight. King Arthur did not know that around 1500 years later, the computer could help him estimate the probability of victory or defeat of each knight before each battle.

2. Requirement

In this assignment, students will be provided with a file containing input data, including parameters for a duel between a Round Table knight and a Saxon warrior. The program will calculate and print out the screen the probability of winning the Round Table knight. Students will also be provided with a sample program to read data from the input file and export the results to the screen.

3. Input

The program input is stored in a file named *input.txt*. This file will keep information about the Knights of the Round Table and the Saxon warriors participating in the duel, whose structure is as follows.



where:

- *baseHP$_1$*: The health point of the Knights of the Round Table, which is an integer between 99 and 999
- *wp$_1$*: information about the weapons of the Knights of the Round Table, which in an integer of the values 0,1,2,3.
- *baseHP$_2$*: The health point of the Saxon warrior, which is an integer between 88 and 888
- *wp$_2$*: information about the weapons of the Saxon warrior, which in an integer of the values 0,1,2,3
- *ground*: Information about the place of battlefiled, which is an integer between 1 and 999

4. Output

The program will output the probability value *p(R)* directly into the screen to predict the possibility of the Round Table knight's victory in hand-to-hand combat. The value of *p(R)* will be calculated by the following formula:

$$p(R) = \frac{realHP_1 - realHP_2 + 999}{2000}$$

where *realHP1* and *realHP*2 are real strength points of the Knight and the Saxon warrior when fighting. This strength point will be calculated based on the health point and weapons used, described as follows.

i) $wp_i = 1$: Normal weapons are used, then $realHP_i = baseHP_i$

ii) $wp_i = 0$: The fighter forgot his and must fight with his bare hands. Then $realHP_i = baseHP_i$ /10 (rounded).

In addition, fighters will gain additional advantage when fighting on familiar battlefield, described as follows.

iii) If $ground = baseHP_i$, then $realHP_i$ will be increased by 10% after calculating the weapon points as described in articles i and ii. However, if $realHP_i$ exceeds 999, it will be automatically reduced to 999

The value printed on the screen will take precisely the form of **0.XX** (that is, the decimal part has exactly two digits). DO NOT PRINT MORE ANY OTHER INFORMATION ON THE SCREEN (including the carriage return). Students are encouraged to use the built-in *display()* function in the init code of *arthur.cpp* (see more in Section 5) to display data to the screen.

Example 1. If $baseHP_1 = 450$, $wp_1 = 1$, $baseHP_2 = 150$, $wp_2 = 1$, $ground = 302$, output is (450-150+999)/2000 = **0.65**.

Example 2. If $baseHP_1 = 807$, $wp_1 = 0$, $baseHP_2 = 750$, $wp_2 = 1$, $ground = 156$, output is (80-750+999)/2000 = **0.16**.

Example 3. If $baseHP_1 = 417$, $wp_1 = 1$, $baseHP_2 = 416$, $wp_2 = 0$, $ground = 417$, output (417*1.1-41+999)/2000 = **0.71**.

Example 4. If $baseHP_1$ = 235, wp_1= 1, $baseHP_2$ = 624, wp_2= 0, *ground* = 624, output is (235-62*1.1+999)/2000 = **0.58**.

Example 5. If $baseHP_1$ = 998, wp_1= 1, $baseHP_2$ = 517, wp_2= 1, *ground* = 998, output is (999-517+999)/2000 = **0.74**.

In addition, the program also considers some special data cases as follows:

iv) If wp_i =2, the weapon used is a special armor made from *mithril*. No weapon can penetrate this armor, so the fighter wearing this armor will never lose. In this case the fighter's $realHP_i$ is still calculated using the descriptions in articles *i* and *iii* and the result will be calculated as same; however, if $realHP_i$ is lower than the opponent's one, the output will be 0.50 (draw match)

Example 6. If $baseHP_1$ = 238, wp_1= 2, $baseHP_2$ = 114, wp_2= 1, *ground* = 145, output is (238-114+999)/2000 = **0.56**.

Example 7. If $baseHP_1$ = 738, wp_1= 1, $baseHP_2$ = 45, wp_2= 2, *ground* = 26, output is **0.50**.

v) If wp_i = 3, the weapon used is the Excalibu swordr. For a Saxon warrior, this sword is just a normal weapon, so $realHP_2$ is still calculated normally using the description in i and iii. If a Knight of the Round Table uses this sword, the $realHP_1$ will be doubled after calculation as described in articles i and iii. However, if the $realHP_1$ value after calculation is higher than 999, it will automatically be reduced to 999. Note that when a Round Table knight uses the Excalibur sword, if the opponent has *mithril* armor, this armor will also lose its magic, becoming a normal armor.

Example 8. If $baseHP_1$ = 414, wp_1= 1, $baseHP_2$ = 415, wp_2= 3, *ground* = 199, output is (414-415+999)/2000 = **0.50**.

Example 9. If $baseHP_1$ = 221, wp_1= 3, $baseHP_2$ = 600, wp_2= 1, *ground* = 221, output is (221*1.1*2-600+999)/2000 = **0.44**.

Example 10. If $baseHP_1$ = 612, wp_1= 3, $baseHP_2$ = 800, wp_2= 1, *ground* = 800, output is (999-800*1.1+999)/2000 = **0.56**.

Example 11. If $baseHP_1$ = 189, wp_1= 3, $baseHP_2$ = 517, wp_2= 2, *ground* = 444, output is (189*2 – 517 + 999)/2000 = **0.43**.

vi) If $baseHP_1$ = 999, King Arthur himself went to the battle. As a result, Arthur always wins regardless of who the opponent is and what weapon he uses. The result will be printed on the screen as **1**. (Not in **0.XX** format as usual)

vii) If $baseHP_2$ = 888, Cerdic himself went to the battle. No Knights of the Round Table matches Cerdic, no matter what weapons they used. In this case, the result on the screen will be 0.00. However, if Cerdic's opponent is King Arthur, the result will be processed as described in article vi.

viii) (*Bonus*) If $baseHP_i$ is a prime number, the corresponding fighter is actually a disguised Paladin. Because Paladin has the ability to use magic in battle, he will certainly win no matter what weapon he uses. Paladins only lose when they encounter Arthur, Cerdic or a Paladin with a higher *baseHP*. If the winning Paladin is a Round Table knight, the screen will print the value 0.99; If the winning Paladin is a Saxon warrior, the screen will print the value 0.01. In case the two Paladins have equal *baseHP*, the screen will print the value 0.50. In case a Paladin meets Arthur or Cerdic, the result will be printed as described in vi and vii respectively.

5. Initialization

Students download the *Assigment_0.zip* file from the course's Web site. When extracting this file, you will get the following files.

`input.txt`	An input example
`arthur.cpp`	Initial code
`Assignmen_0.pdf`	Assignment description

The file *input.txt* is a sample input file as described in Section 4. File *arthur.cpp* is an initialization program, in which the *readFile()* and *display()* functions have been developed already. Students are strongly encouraged to use these functions.

6. Submission

The deadline for submission is **23:55 on Friday, April 17th, 2020**. Students must use the account on the BKeL system to submit your work. We DO NOT receive any thing related to assignment which is sent via email or any other form. Late submissions will NOT be accepted.

In addition to the libraries already used in *arthur.cpp,* students should NOT use any other library. When submitting the result, students must use the account in BKeL. Students only submit one file *arthur.cpp*. The file submitted must be the original program file, BUT NOT A COMPRESS FILE (ZIP) WHEN SUBMITTING. **Students must check their program on Cygwin before submitting.**

7. Plagiarism

Students must do the assignment themselves. Their works will be considered plagiarism if:

- There is large similarity between the source code of submissions. In this case, all submissions are considered plagiarism. Therefore, students must protect the source code.
- Students do not understand the source code written by themselves, except for the parts of code provided in the initialization program. Students can consult from any source, but make sure that they understand the meaning of all the code they write.

In the case of a conclusion of cheating, students will receive a zero for the entire subject (not just the assignment).

DO NOT ACCEPT ANY EXCUSE AND NO EXCEPTIONS!

After each assignment is submitted, there will be a number of students who are randomly interviewed to prove that the assignment has been submitted by themselves.