CS201 – SPRING 2020-2021
Take-Home Exam 3
– The Sorting Hat –
Due 5th May, Wednesday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on file streams.

Your take-home exams will be automatically graded using GradeChecker, so it is very important to satisfy the exact same output given in the sample runs. You can utilize GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the take-home exam (**all** .cpp and .txt files for this take-home exam). Additionally, you should submit all of your files to SUCourse **without zipping** them. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_THEnumber.cpp" (all lowercase letters). Please check the submission procedures of the take-home exam, which are listed at the end of this document.

To get help using GradeChecker you may ask questions to the list of your grader TAs: cs201gchelp@lists.sabanciuniv.edu.

Description

In this homework assignment, we will implement the famous "sorting hat" quiz in the Harry Potter world. In one file, you are given the users' answers to a sorting hat quiz, where each answer has a specific score (there are no correct answers, rather all of the answers have certain weights). These weights are stored in a separate file, in a manner that all answers to the questions are written in order of their points. The first word on the line is 1 point, the second is 2 points and so on.

Inputs and program flow

The program should start with displaying the welcome message and then entering the file names for the answer key file and for the students' answers. These file names should be checked for correctness, until the correct file name is entered. After

opening the files, you should start processing the answer file. The answer file format is the following;

```
Harry Potter Owl Phoenix Weak Expelliarmus Friendship Seeker
```

First two words are reserved for the user's name and lastname. After these, there are six answers to six questions. Each line in the answer key file corresponds to one answer. Take, for example the first line in the answer key file:

```
Toad Owl Cat None
```

Here, Toad is 1 point, Owl is 2 points, Cat is 3 points and None is 4 points. Since the user's answer matches with Owl, they get 2 points from this answer. Remember to NOT hardcode these answers to your program since the questions & answers can change in the grading. This processing should be made for all of the answers. After the score is calculated by adding up the points for each answer, you should display the user's name, their score, and their Hogwarts house.

Example answers.txt

```
Harry Potter Owl Phoenix Weak Expelliarmus Friendship Seeker
Draco Malfoy None Unicorn Weak Crucio Diligence Seeker
```

Example answerKey.txt

```
Toad Owl Cat None
Phoenix Dragon Unicorn
Boring Unkind Ignorant Weak
Protego Stupefy Expelliarmus Crucio
Friendship Learning Concealment Diligence
None Chaser Keeper Seeker
```

Each row in the answer key file is the grading criterion of the corresponding question. Row 1 is for the first question, row 2 is for the second question, etc. Each answer consists of a single word. Between each word in any document there will be at least one space character, however there might be any number of spaces, too. As the sorting hat, you must gather the information from the answers of each student (name, surname, 6 answers) and compare it with possible answers within the answerKey document. You will repeat this process until the answers file contains no more student information. Alse be aware that the answers file can contain empty lines. You should handle that too.

NOTE: You cannot use arrays or vectors in this homework.

Each row in each document has its purpose and order within each line. The answers file will contain **exactly one** name and **exactly one** surname and **exactly 6** answer words. The keys file will contain **exactly 6** lines/rows and each row will contain **at least one** answer word. Depending on the position of the word, you must decide on its point value.

The thresholds for the houses is as follows:

score < 10	Hufflepuff
10 <= score < 15	Ravenclaw
15 <= score < 21	Gryffindor
21 <= score	Slytherin

Please refer to the "Sample Runs" section for some examples and further details.

IMPORTANT!

If your code does not compile, then you will get **zero**. Please be careful about this and double check your code before submission.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the mentioned format.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some points in the best scenario.

No abrupt program termination please!

Especially during the input check, you may want to stop the execution of the program at a specific place in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

Grade Checker Note

This take home exam requires you to upload the .txt files along with your source code. To get 100, you must upload all of the .txt files (answers.txt, answers2.txt, answerKey.txt and your main.cpp) to GradeChecker. This is not a submission guide for SuCourse+.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You should follow the input order in these examples and the prompts that your program will display **must** be **exactly the same** as given in the following examples.

Sample Run 1

```
Hello and welcome to the Hogwarts house sorting hat program!
When I call your name, you'll come forth, I shall place the sorting hat
on your head, and you will be sorted into your houses.
Enter the name of the key file: answerKey.txt
Enter the name of the answer file: answers1.txt
Oh! we have Harry Potter under the sorting hat. Harry, you scored 15
points, you are a Gryffindor!
Oh! we have Cedric Diggory under the sorting hat. Cedric, you scored 8
points, you are a Hufflepuff!
Oh! we have Luna Lovegood under the sorting hat. Luna, you scored 11
points, you are a Ravenclaw!
Oh! we have Draco Malfoy under the sorting hat. Draco, you scored 23
points, you are a Slytherin!
Oh! we have Roger Davies under the sorting hat. Roger, you scored 13
points, you are a Ravenclaw!
```

Sample Run 2

```
Hello and welcome to the Hogwarts house sorting hat program!
When I call your name, you'll come forth, I shall place the sorting hat on your head, and you will be sorted into your houses.
Enter the name of the key file: keys.txt
Error: Cannot open file keys.txt
Enter the name of the key file: answerKey.txt
Enter the name of the answer file: answers2.txt
Oh! we have Hermione Granger under the sorting hat. Hermione, you scored 14 points, you are a Ravenclaw!
Oh! we have Oliver Wood under the sorting hat. Oliver, you scored 15 points, you are a Gryffindor!
Oh! we have Cho Chang under the sorting hat. Cho, you scored 10 points, you are a Ravenclaw!
Oh! we have Pansy Parkinson under the sorting hat. Pansy, you scored 21 points, you are a Slytherin!
Oh! we have Susan Bones under the sorting hat. Susan, you scored 15 points, you are a Gryffindor!
```

Sample Run 3

```
Hello and welcome to the Hogwarts house sorting hat program!
When I call your name, you'll come forth, I shall place the sorting hat on your head, and you will be sorted into your houses.
Enter the name of the key file: answerKey.txt
Enter the name of the answer file: answer.txt
Error: Cannot open file answer.txt
Enter the name of the answer file: answers2
Error: Cannot open file answers2
Enter the name of the answer file: answers2.txt
Oh! we have Hermione Granger under the sorting hat. Hermione, you scored 14 points, you are a Ravenclaw!
Oh! we have Oliver Wood under the sorting hat. Oliver, you scored 15 points, you are a Gryffindor!
Oh! we have Cho Chang under the sorting hat. Cho, you scored 10 points, you are a Ravenclaw!
Oh! we have Pansy Parkinson under the sorting hat. Pansy, you scored 21 points, you are a Slytherin!
Oh! we have Susan Bones under the sorting hat. Susan, you scored 15 points, you are a Gryffindor!
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all take-home exams, unless otherwise noted.

- **How to get help?**

You can use GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the course website.

- **What and Where to Submit**

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using Xcode (macOS users).

It'd be a good idea to write your name and last name in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "Barış Altop", and if you want to write it as comment; then you must type it as follows:
// Baris Altop

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:
    - Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
    - Name your cpp file that contains your program as follows: "**SUCourseUsername_THEnumber.cpp**"
    - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**). For example, if your SUCourse username is "**altop**", then the file name should be: **altop_the1*.cpp*** (please only use lowercase letters).

    - Do not add any other character or phrase to the file name.

- Please make sure that this file is the latest version of your take-home exam program.

- Submit your work **through SUCourse only**! You can use GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

- Grading, Review and Objections

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your take-home exam, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.

- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.

- Please submit your **own** work only. It is really easy to find "similar" programs!

- Plagiarism will not be tolerated. Please check our plagiarism policy given in the Syllabus or on the course website.

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your take-home exam from the email address provided in the comment section of your announced take-home exam grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the take-home exam tab to see the problem with your take-home exam.
- Download the file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Şevval Şimşek & Gülşen Demiröz & Barış Altop