CS201 – SPRING 2020-2021
Take-Home Exam 2
– Hangman Game –
Due 11 April, Sunday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on strings and loops.

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 (**only** *your_main_cpp* file for this take-home exam). Additionally, you should submit all of your files to SUCourse (**only** *your_main_cpp* file for this take-home exam) **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 Take-Home Exam, you will write a C++ program that simulates the hangman game. There are two players, player one chooses a word which is secret, and player two tries to guess the word letter by letter or as a whole. Since you are now familiar with input checks, all of the inputs must be checked to make sure they are valid.

The inputs and the flow is as follows:

First, the players enter their names, and after this point they are referred by their names. After players are welcomed to the game, player one enters the secret word that player two will try to guess. This word is checked so that every character is an alphabetical character. Both uppercase and lowercase letters are accepted, however you should convert uppercase characters to lowercase for ease of use. For the

lowercase conversion, **you have to use the strutils library**, which we will provide along with the THE document (strutils.cpp and strutils.h). You cannot use any other library for this conversion and will lose points, and keep in mind that you need to add these files to GradeChecker to run your code. After making sure the entered word is in the correct format, you should create a secret word in order to display it to the second player, in the form of **"------"** where each **"-"** character represents a letter in the original word. As correct guesses are made, you should replace the **"-"** characters with the actual characters of the word.

Player two has a total of 6 lives, in which the first guess is free and the last five lives represent the hangman's two arms, two legs and one head. After each incorrect guess, you should display appropriate messages. After the first incorrect guess, display the message **"You have no free guess left.".** If the free life is lost, after each incorrect guess, display a message saying **"You have lost a leg"** or **"You have lost an arm"** or **"You have lost a head"**. The hangman has two legs, two arms and one head, and if the hangman loses the head, the game is over, meaning player two loses the game.

When entering a guess, player two has two options, they can either enter the whole word (option1) or enter a letter (option2). After this option is entered, a word or a letter is inputted by the player two and you should again check if the entered guess is correct in terms of format (Same rules as the original word). For the letter option, the entered string must only consist of one letter, and if it is longer, the guess is not accepted. In the case of all invalid entries, you should display the **"Invalid entry! Try again."** message. If the guess is just a character, again make sure to convert it to lowercase, and then check if this guess is made before. If it is entered before, again display an error message saying **"Guess entered before! Try again.".** If this is a new character that hasn't been entered before, then you should proceed to check if this guessed character belongs to the secret word, and display the proper outputs. **You are not allowed to use arrays, vectors or maps for this THE.**

If the player enters a word, it should match exactly to the secret word to be a correct guess, meaning we do not accept partial matches. If there's a full match, player 2 wins the game and you should display a message saying **"Congratulations** *player2***, you won!"** by replacing the *player2* with their name. Again here, uppercase letters are accepted but you should convert the characters to lowercase. If the word does not match, player 2 loses a life.

After each guess, you should display the redacted secret, and ask player two if they want to enter a word or a letter again, until the game is over. The game over condition is 1) Player two uses up all the lives or 2) Player two makes a correct guess by either entering a whole word or entering all characters correctly.

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.

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

```
Welcome to the HANGMAN GAME
```

```
--------------------------
Player one, please enter your name: Gulsen
Player two, please enter your name: Baris
OK Gulsen and Baris. Let's start the game!
Gulsen please input the word you want Baris to guess: cs201
Invalid word! Try again.
Gulsen, please input the word you want Baris to guess: whatislove?
Invalid word! Try again.
Gulsen, please input the word you want Baris to guess: apple
Baris, you have 1 free guess, after that you will lose limbs!
The word is: -----
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: 5
Invalid entry! Try again.
The word is: -----
Baris, do you want to guess the word(1) or guess a letter(2)? *
Invalid option!
The word is: -----
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: A
The word is: a----
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: pl
Invalid entry! Try again.
The word is: a----
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: p
The word is: app--
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: e
The word is: app-e
Baris, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: l
The word is: apple
Congratulations Baris, you won!
Baris won the game!
```

Sample Run 2

Welcome to the HANGMAN GAME

Player one, please enter your name: ***Berfin***
Player two, please enter your name: ***Defne***
OK Berfin and Defne. Let's start the game!
Berfin please input the word you want Defne to guess: ***FroZeN***
Defne, you have 1 free guess, after that you will lose limbs!
The word is: ------
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***a***
You have no free guess left.
The word is: ------
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***e***
The word is: ----e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***a***
Guess entered before! Try again.
The word is: ----e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***o***
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***1***
Your guess: ***model***
You have lost a leg!
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***1***
Your guess: ***hower***
You have lost a leg!
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***1***
Your guess: ***2joke***r
Invalid entry! Try again.
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***m***
You have lost an arm!
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? ***2***
Your guess: ***p***

```
You have lost an arm!
The word is: --o-e-
Defne, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: t
You have lost a head! GAME OVER!
Defne lost the game :(
```

Sample Run 3

```
Welcome to the HANGMAN GAME
---------------------------
Player one, please enter your name: Nasim
Player two, please enter your name: Ufuk
OK Nasim and Ufuk. Let's start the game!
Nasim please input the word you want Ufuk to guess: raBBit
Ufuk, you have 1 free guess, after that you will lose limbs!
The word is: ------
Ufuk, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: a
The word is: -a----
Ufuk, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: i
The word is: -a--i-
Ufuk, do you want to guess the word(1) or guess a letter(2)? 2
Your guess: b
The word is: -abbi-
Ufuk, do you want to guess the word(1) or guess a letter(2)? 1
Your guess: rabb1t
Invalid entry! Try again.
The word is: -abbi-
Ufuk, do you want to guess the word(1) or guess a letter(2)? 1
Your guess: RABBIT
The word is: rabbit
Congratulations Ufuk, you won!
Ufuk won the game!
```

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