

CS201 – Spring 2017-2018 - Sabancı University

Homework #6: A Simple Betting Program

Due May 10, Thursday, 22:00

Warning

We will be automatically grading your following homeworks as GradeChecker does, so it is very important to satisfy the exact same output with the sample runs. You can utilize GradeChecker (<http://sky.sabanciuniv.edu:8080/GradeChecker/>) to check whether your code is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the homework (your_main_cpp file, strutils.h, strutils.cpp, and all txt files). Additionally, you should submit all of your files to SUCourse (your main cpp file, strutils.h, strutils.cpp and all txt files) **without zipping them. Name of your main cpp file should be in the expected format** ("sucourseusername_lastname_name_hwnumber.cpp", check the submission procedure of the homework).

Description

In this homework, you will implement a simple legal betting system like Iddaa. Every week, Iddaa announces a program which contains a set of matches with the odds of every possible result. Then, a user guesses the result of at least 3 matches and pays some money (called as *Stake*) for this coupon. If this user knows all the results correctly, then he/she wins some money depending on the odds of the results and how much money has been put as stake. Otherwise, the user will not make any money and the stake will be lost.

If a user guesses n matches and he/she knows all the results correctly, then the amount of money earned can be calculated as follows:

$$\text{earnedMoney} = \text{stake} * \text{odd1} * \text{odd2} * \dots * \text{oddn};$$

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 abovementioned format.
- **Prompts before inputs and outputs** must be **exactly the same** with examples.

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

IMPORTANT!

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

Name and Structure of the Input Files

There will be three text files as input files for your program (i.e. your program will read some data from these files) and these files are generated outside of the program using Notepad (we provide example input files within the zip package of this homework).

Name of these input files will be entered by the user of your program. At the beginning of your program, user will be asked to enter an input file name for the whole program of the week. If the user enters a name for a file that does not exist, your program should display an appropriate message and ask for the name of a new input file name until the file is successfully opened. After you got the correct filename for the whole program, then you will ask for the other filenames for the user coupon and match scores until the files are opened successfully.

In the file for the whole program (“program.txt” in the zip package), each line contains information for a specific match in the following format:

MatchCode Team1 – Team2 odd1 odd0 odd2

Each match has a unique *MatchCode* which is an integer. *Team1* is the name of home team and *Team2* is the name of the away team. Both team names may contain more than one word. Note that there is one space, exactly one dash and one more space (“ – ”) between team names. *odd1* represents the odds value if *Team1* beats *Teams2*. *odd0* is the odds value if the match ends with a draw. Lastly, *odd2* is the odds value if *Team2* beats *Team1*. All *odd1*, *odd0* and *odd2* values are double values. You cannot make any assumption about the number of spaces between each data; there might be more than one space between each of them.

In the file for the user coupon (“coupon1.txt” or “coupon2.txt” in the zip package), the first line shows how much money has been put as stake as following:

STAKE: 10.55 TL

Then, each line in this file contains the user guesses in the following format:

MatchCode Guess

MatchCode represents the unique match code (integer) to be guessed by user. *Guess* value might be 0, 1 or 2; as 1 represents home team wins, 2 represents away team wins and 0 represents there will be a draw among teams. You may assume that user will type *Guess* as 0, 1 or 2; so you do not need to check its format. However, *MatchCode* may not exist in the “program.txt” where this case makes this coupon an invalid coupon.

The last input file is for the match scores (“matchscores.txt” in the zip package) and each line in this file represents the real results for each match in following format:

MatchCode MatchResult

MatchResult represents *NumberOfGoalsScoreedByTeam1-NumberOfGoalsScoreedByTeam2* (without any spaces). However there might be any number of spaces between *MatchCode* and *MatchResult*.

No format check for the contents of the input files is needed. You may assume that all data are in the correct format and you may also assume that there is not any empty line in the input files. You may examine the sample input files provided in the zip package for this homework (hw6.zip).

Output

You will display the results on command line screen using `cout`. In the output, each line should contain the result of a guess made by the user in the following format:

MatchCode Team1 – Team2 oddsValueofGuess SUCCESSorFAILED

oddsValueofGuess represents the odd value of user’s guess. *SUCCESSorFAILED* may only be “SUCCESS” or “FAILED” according to the user’s correctness. There should be **one** space between each data. **Note that all these matches should be sorted in the same order with the matches in user coupon.**

If there is a match with the invalid match code in user coupon, then this match should not be printed. In this case, you should print “*INVALID COUPON*” at the end of the output. (See coupon3.txt and Sample Runs)

If there are less than 3 matches in user’s coupon then this coupon will again be an invalid coupon even if all match codes are valid. (See coupon4.txt and Sample Runs)

If there are not any invalid matches, but at least one of the matches has been guessed incorrectly, then you should print how much money this user lost at the end of the output. (See coupon2.txt and Sample Runs)

If there are not any invalid matches and all matches have been guessed correctly, then you should print the total odds value and total earned money. (See coupon1.txt and Sample Runs)

```
totalOddsValue = odd1 * odd2 * ... * oddn;  
earnedMoney    = stake * totalOddsValue;
```

Some common problems/questions and solutions/hints

Question 1: I enter a correct input filename, but my program does not open corresponding files.

Answer 1: If you run your program in the VC++ environment, then the input file must be in the same folder as the .vcxproj file of your project. Otherwise although the file may be available and matching the name, your program may not be able to find and open the file.

Question 2: I formed a loop in order to check if the input file is opened successfully or not. However, after a file name is entered wrong, even if I enter a correct name, it does not accept it.

Answer 2: The reason, most probably, is not clearing the input file stream before trying to re-open it. In order to re-open of file stream object after a failed open you have to call the clear member function on this file stream.

Question 3: My program works fine if I enter the correct file names. However, if I enter a wrong file name first and then the correct file name, the output files are either empty or the results are problematic.

Answer 3: The reason, most probably, is redefinition of the file stream object in the loop that you check whether the files are opened or not. Do not redefine the file streams in the loops. As mentioned in the question above, clear them if needed.

No abrupt program termination please!

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 your program will display must be **exactly the same** as in the following examples.

Sample Run 1

This is a simple betting program.

Please enter a filename which contains whole betting program: ***program.txt***

Please enter a filename which contains the matches played by a user: ***coupon1.txt***

Please enter a filename which contains the match scores of the week: ***matchscores.txt***

283 Bursaspor - Elazigspor 1.6 SUCCESS
284 Galatasaray - Kasimpasa 6.75 SUCCESS
227 Akhisar Bld.G.Spor - Kayserispor 3.2 SUCCESS

TOTAL ODDS: 34.56

You won 364.608 TL

!!! BET "LEGALLY" AND "RESPONSIBLY" !!!

Press any key to continue . . .

Sample Run 2

This is a simple betting program.

Please enter a filename which contains whole betting program: ***program.tx***

Error for opening program.tx Please enter filename again: ***program.txt***

Please enter a filename which contains the matches played by a user: ***coupon2.tx***

Error for opening coupon2.tx Please enter filename again: ***coupon2.txt***

Please enter a filename which contains the match scores of the week: ***matchscores.txt***

283 Bursaspor - Elazigspor 1.6 SUCCESS

284 Galatasaray - Kasimpasa 6.75 SUCCESS

227 Akhisar Bld.G.Spor - Kayserispor 3 FAILED

YOU LOST, 10.55 TL HAS GONE

!!! BET "LEGALLY" AND "RESPONSIBLY" !!!

Press any key to continue . . .

Sample Run 3

This is a simple betting program.

Please enter a filename which contains whole betting program: ***program.txt***

Please enter a filename which contains the matches played by a user: ***coupon3.txt***

Please enter a filename which contains the match scores of the week: ***matchscores.txt***

283 Bursaspor - Elazigspor 1.6 SUCCESS

227 Akhisar Bld.G.Spor - Kayserispor 3.2 SUCCESS

INVALID COUPON

!!! BET "LEGALLY" AND "RESPONSIBLY" !!!

Press any key to continue . . .

Sample Run 4

This is a simple betting program.

Please enter a filename which contains whole betting program: ***program.txt***

Please enter a filename which contains the matches played by a user: ***coupon4.txt***

Please enter a filename which contains the match scores of the week: ***matchscores.txt***

283 Bursaspor - Elazigspor 1.6 SUCCESS

284 Galatasaray - Kasimpasa 6.75 SUCCESS

INVALID COUPON

!!! BET "LEGALLY" AND "RESPONSIBLY" !!!

Press any key to continue . . .

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homeworks, unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS201. Office hours of TAs are at the class website. Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Careful about the full-automatic grading: Your programs will be graded using an automated system. Therefore you should follow the guidelines about input and output order; moreover you should also use same prompts as given in the Sample Runs. Otherwise automated grading process will fail for your homework, and you may get a zero, or in the best scenario you will lose points.

Grading:

- ☐ Late penalty is 10% off of the full grade and only one late day is allowed.
- ☐ **Having a correct program is necessary, but not sufficient to get the full grade.**
Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily

long program (which is bad) and unnecessary code duplications (which is also bad) will also affect your grade.

- ☐ Please submit your own work only (even if it is not working). It is really easy to find out “similar” programs!
- ☐ For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/su_current_courses/cs-201-spring-2008/plagiarism/ and keep in mind that

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 homework from the email address provided in the announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the files 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.

What and where to submit (IMPORTANT)

Submissions guidelines are below. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course). Do not use Turkish characters anywhere in your program (not even in the comments).

Name your submission file:

- ☐ Use only English alphabet lowercase 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_lastname_name_hwnumber.cpp**”
- ☐ Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsizkodyazaroglu, then the file name must be:

cago_ozbugsizkodyazaroglu_caglayan_hw6.cpp

- ☐ Do not add any other character or phrase to the file name.
- ☐ Make sure that this file is the latest version of your homework program.

- ❑ You have to submit all files that are used in your homework (ago_ozbugsizkodyazaroglu_caglayan_hw6.cpp, strutils.h, strutils.cpp, and all txt files) **without zipping them.**

Submission:

- ❑ Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
 - 1) Click on "Assignments" at CS201 SUCourse (not the CS201 web site).
 - 2) Click Homework 6 in the assignments list.
 - 3) Click on "Add Attachments" button.
 - 4) Click on "Browse" button and select all file used in homework.
 - 5) Now, you have to see your file(s) in the "Items to attach" list.
 - 6) Click on "Continue" button.
 - 7) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- ❑ After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
 - 1) Click on "Assignments" at CS201 SUCourse.
 - 2) Click Homework 6 in the assignments list.
 - 3) Click on "Re-submit" button.
 - 4) Click on "Add/remove Attachments" button
 - 5) Remove the existing files by clicking on "remove" link. This step is very important. If you do not delete the old files, we receive both files and the old one may be graded.
 - 6) Click on "Browse" button and select the new files that you want to resubmit.
 - 7) Now, you have to see your new files file in the "Items to attach" list.
 - 8) Click on "Continue" button.
 - 9) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

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

Good Luck!

Ece Egemen, İnanç Arın and Gülşen Demiröz