Getting Started

# Rules of the game

This game is like hand cricket and runs on Python. To know more about hand cricket, visit <https://www.instructables.com/id/How-to-Play-Hand-Cricket/>

However, you will find some differences in the game. The ‘stok’ moves which otherwise adds the bowler’s input to your score adds zero in the app.

Whatever you show your opponent, enter its value. For example, showing an open palm to your opponent represents 5. Input 5 and not ‘open palm’.

The ultimate goal is simple: Score more than your opponent to win.

# OS Pre-requisites

To play this game, you need to have Python software installed. You can download the software from [https://www.python.org/downloads/](https://www.python.org/downloads/release/python-374/)release/python-374 and install it. The software was built on Python 3.7.4, while the newest release as of 27 December 2020 is Python 3.9.1. Go to <https://www.python.org/downloads> to install the latest version.

**NOTE: THIS GAME IS CROSS-PLATFORM COMPATIBLE, FREE AND OPEN SOURCE.**

Supported platforms:

* Microsoft Windows 7 or later
* Android API 24 or later (Apps are available on the Play Store)
* Mac OS X 10.6 or later
* Linux

NOTE: As Windows XP uses Python 2.7 as its last version, developers can modify the code for this purpose.

Disk space: 72 KB (minimum).

# Game pre-requisites:

For the game to work correctly, you’ll need the following Python Files:

* (21.3 KB)
* (690 bytes)
* (1.57 KB)
* (20.4 KB)

Size of game: 44 KB.

# What’s this project?

This is the legacy hand cricket game – implemented in Python and supporting text files.

This project was built entirely on Python 3.7.4 and is compatible with all later versions as well.

The game can be played even without an internet connection.

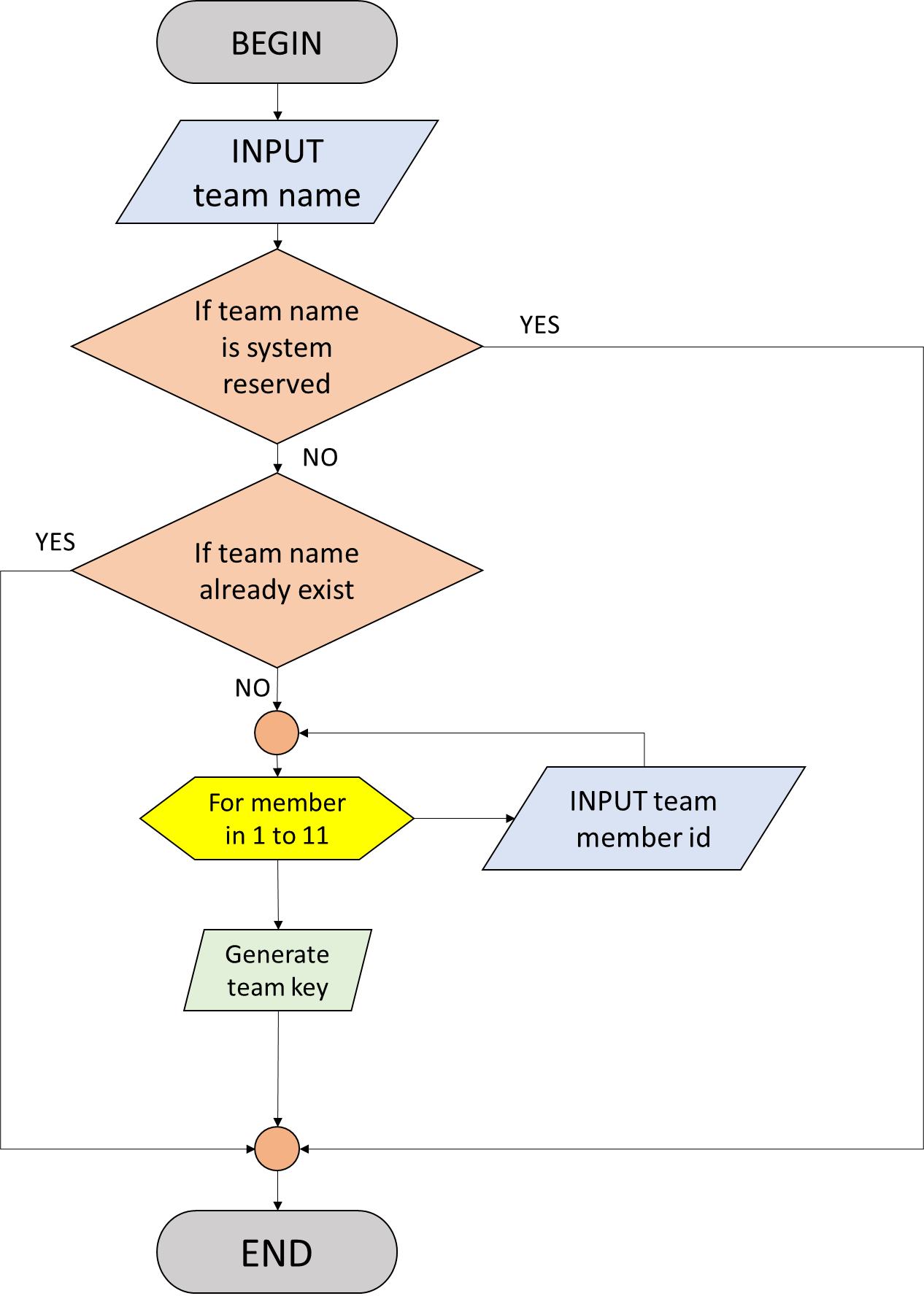
Note that tournament mode is not included in this game to discourage video game addiction. Customized tournaments can be created using individual team files. For such tournaments, at least 1 MB of disk space is recommended.

# Known bugs:

A hack exists wherein a team may abruptly and deliberately close the application to avoid losing a match. This bug is not fixed since a more serious bug, in which unexpected crashes would lead to a loss irrespective of the position, would otherwise overshadow the possibility of victory. There is no provision for rain delay and abandoned matches here.

# Procedure

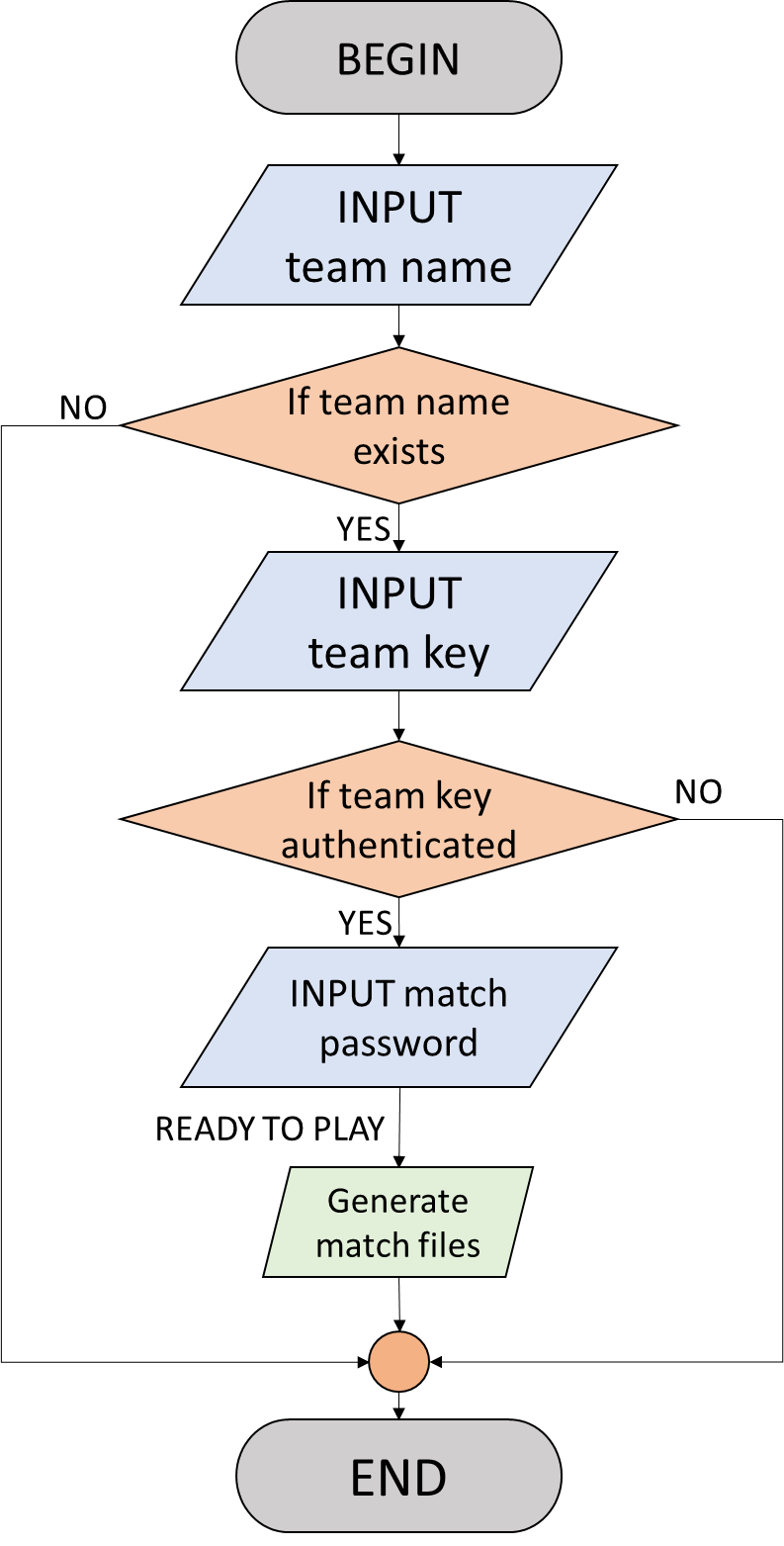
## Setting up a team



NOTE: In the actual program, instead of executing the for loop, the team member ID of all 11 players are input and registered sequentially.

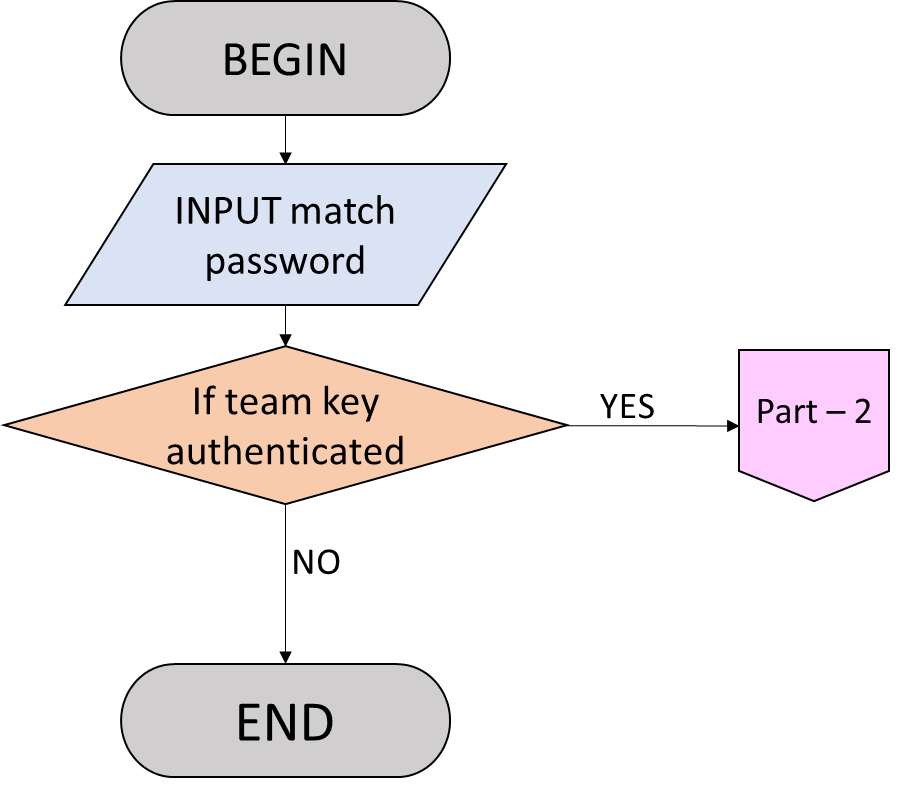
NOTE: If team name already exists, the program will terminate. Reopen the program and try again.

## Registering for a match

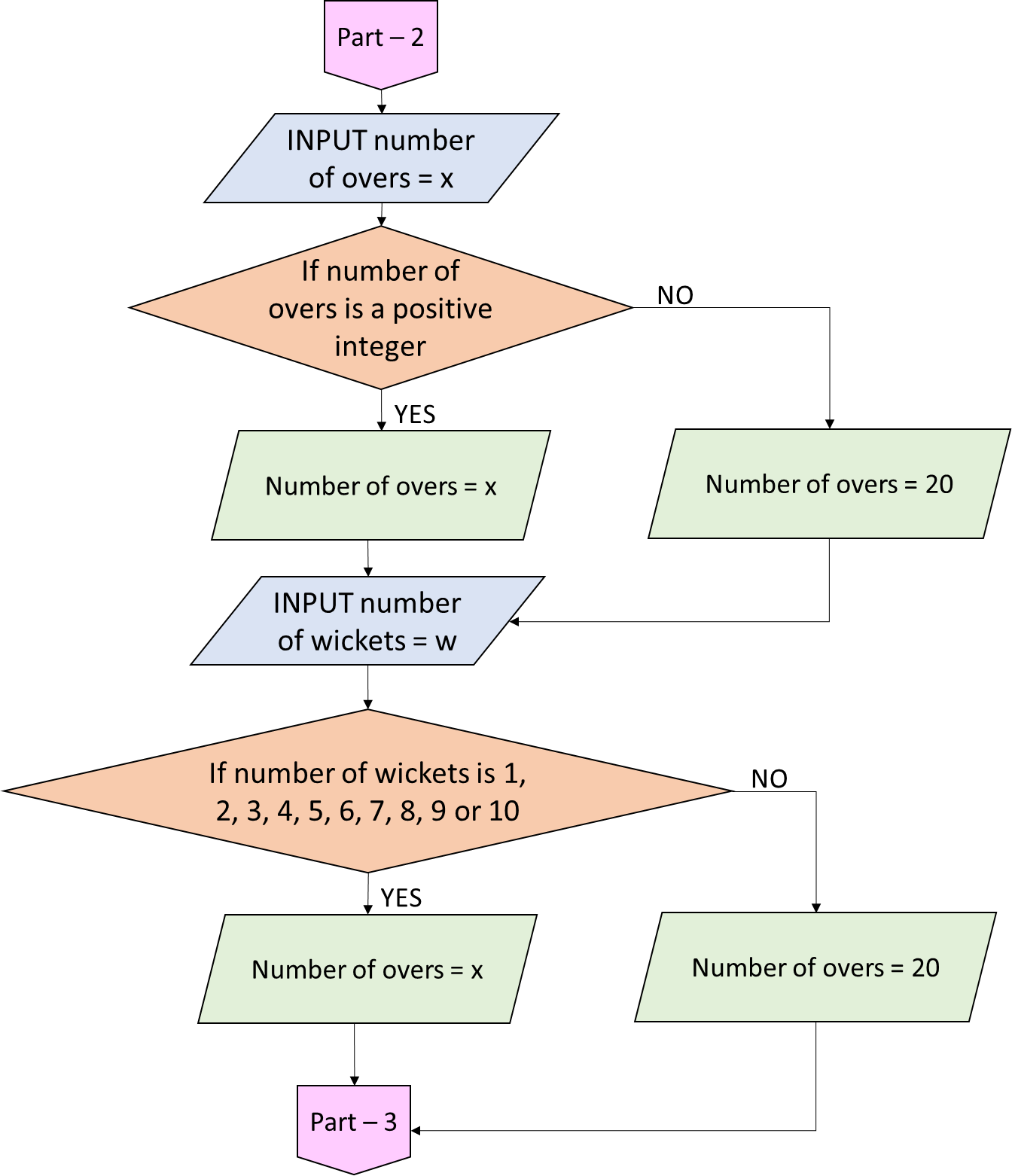


## Gameplay

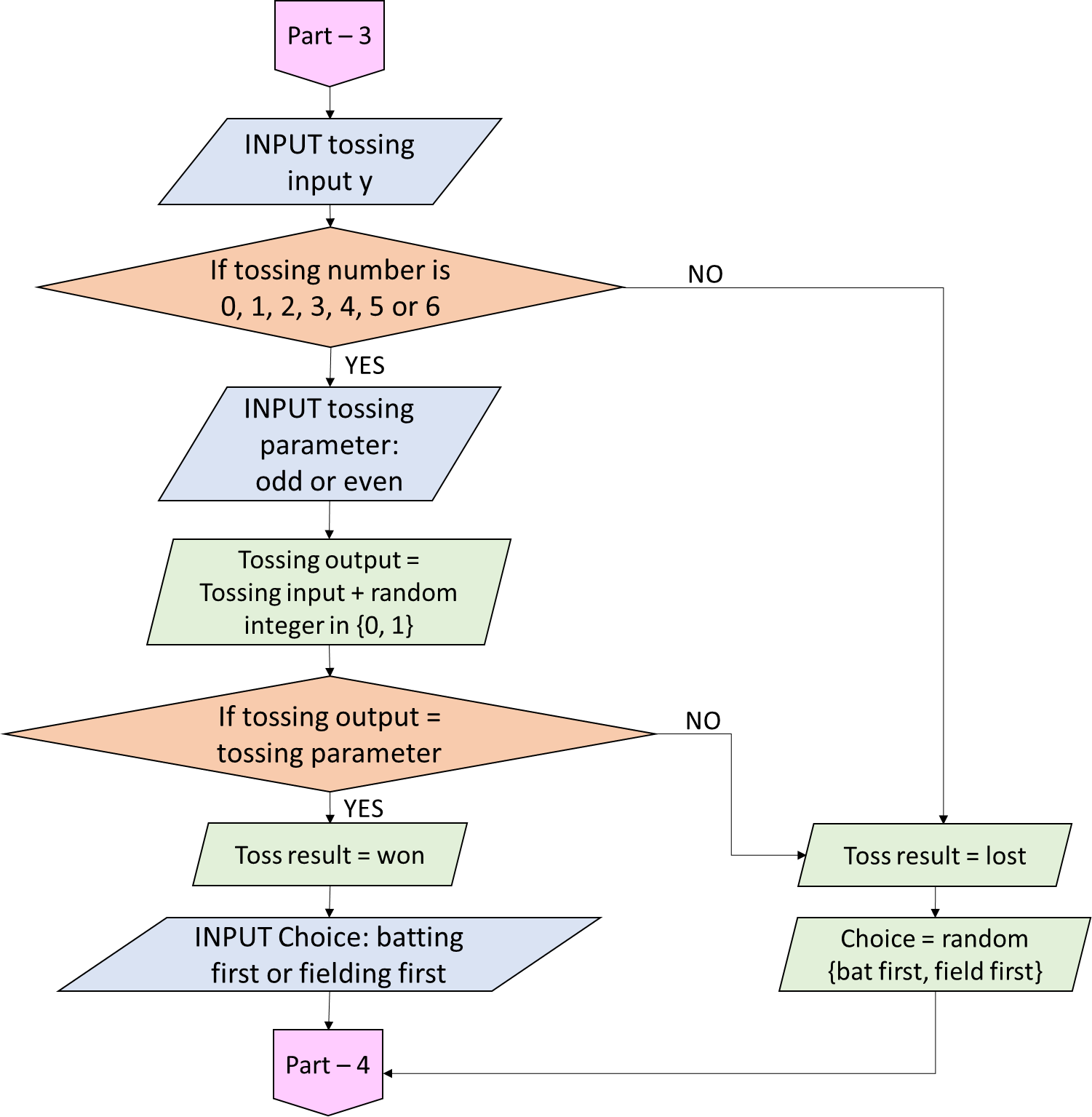
### (Part – 1) Authenticating playing team



### (Part – 2) Deciding number of overs and wickets

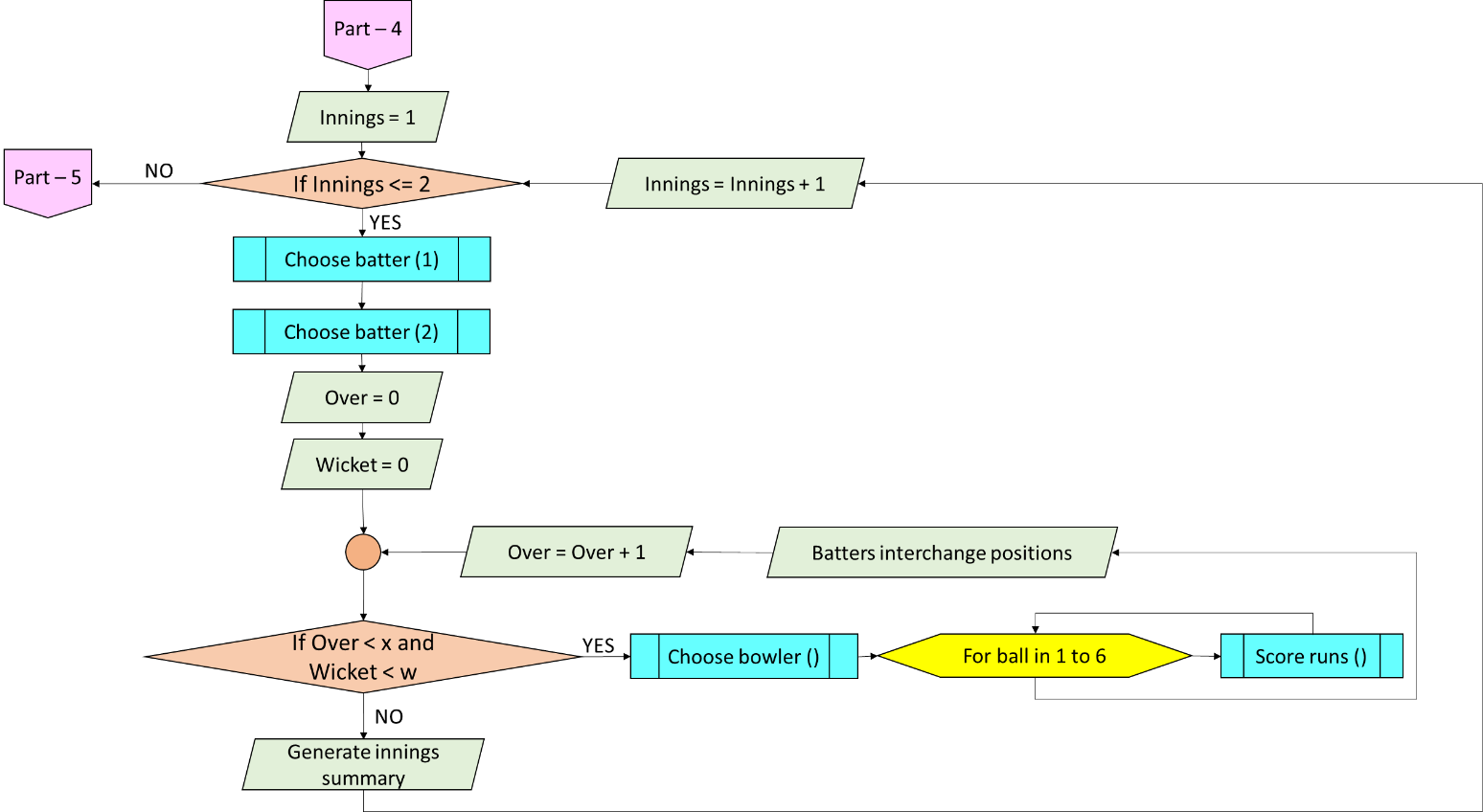


### Toss



### Innings

#### Main function

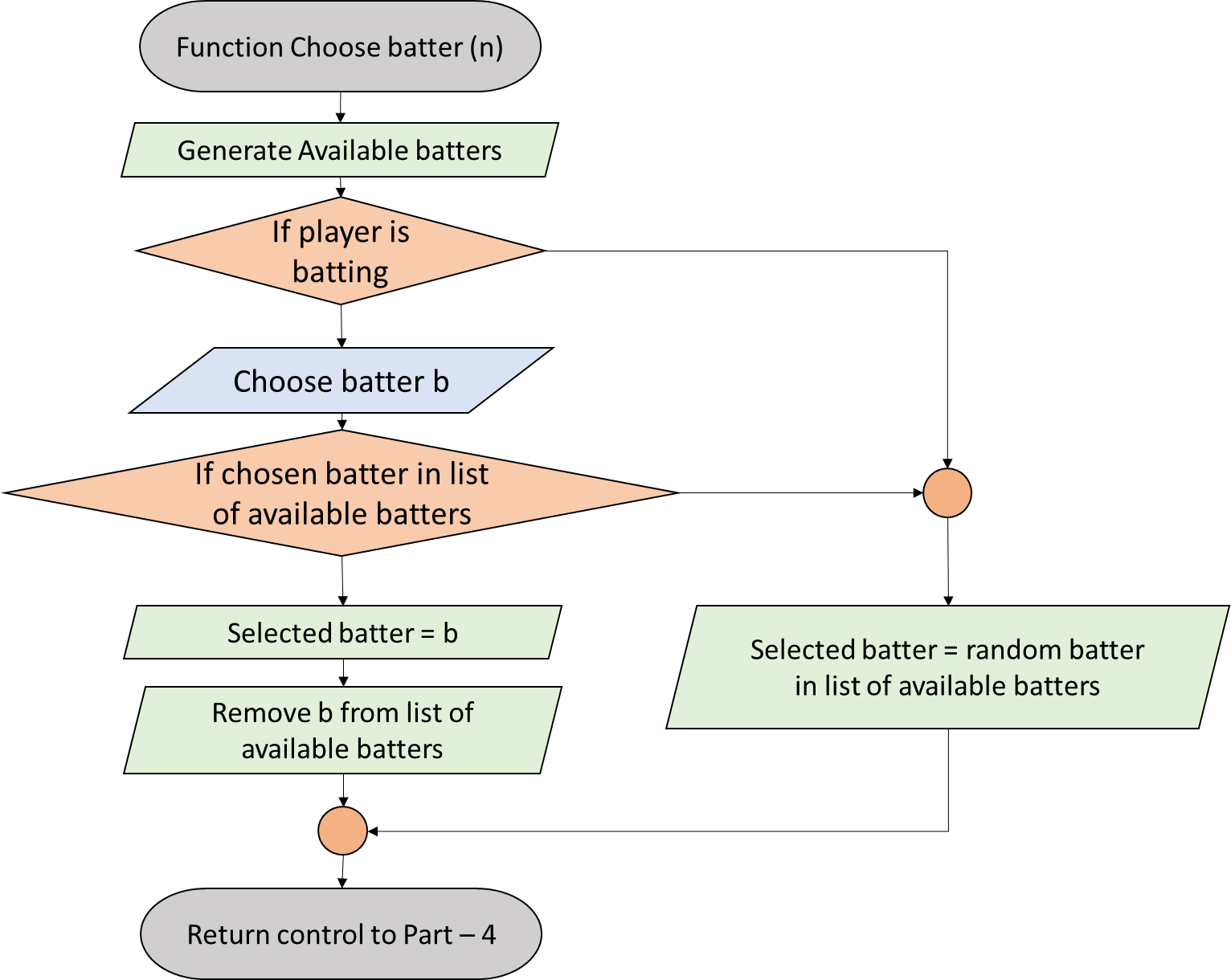


Here, x represents the total number of overs and w represents the total number of wickets.

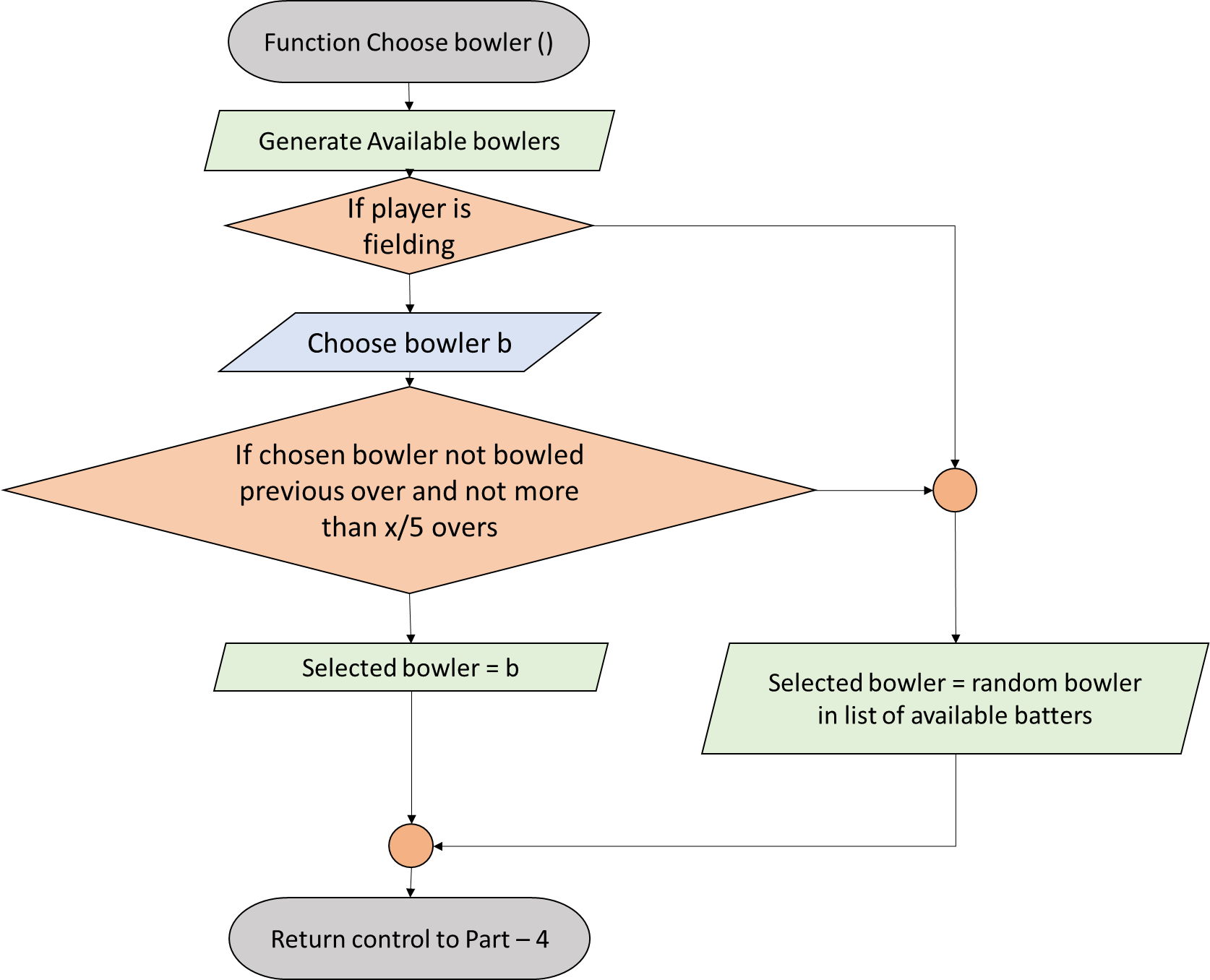
Note that if all the wickets fall, the innings ends irrespective of whether the over was completed or not.

In the super over, only 1 over and 2 wickets per innings is permitted as per official rules.

#### Choose batter (n)



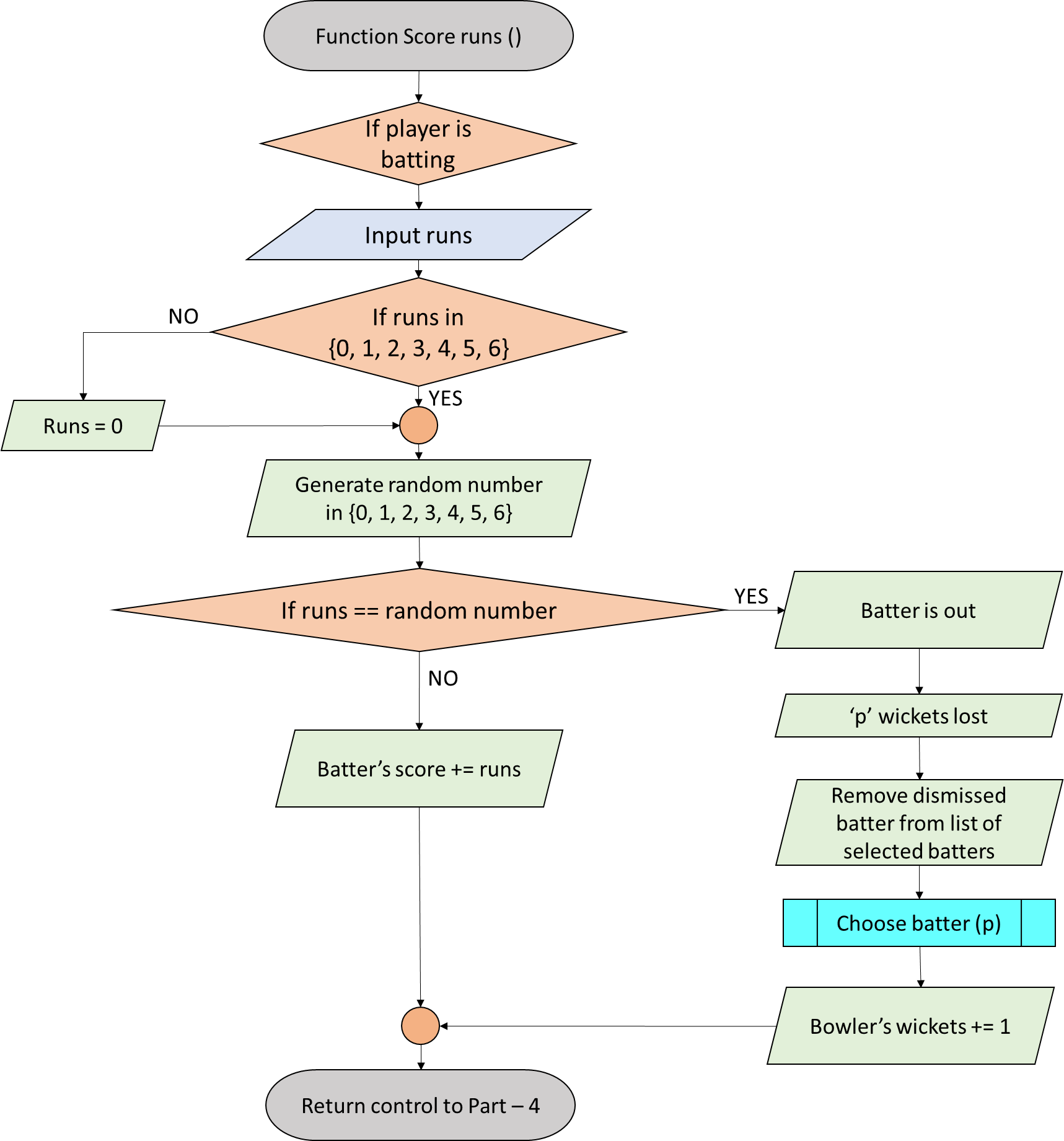
#### Choose bowler



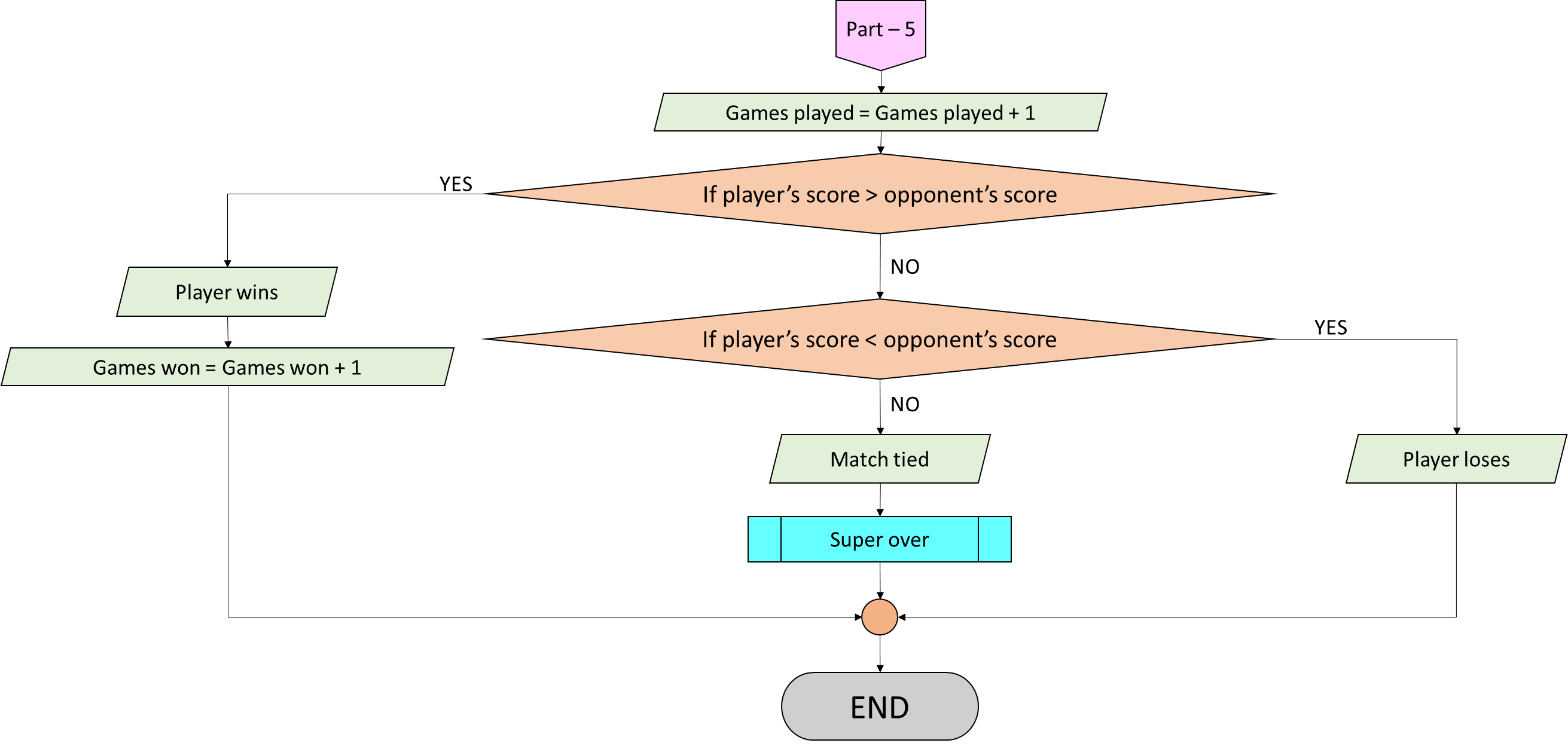
Here, x represents the total number of overs. The condition on x is implemented as per the official rules of cricket.

In the super over, only one bowler can bowl.

#### Score runs



#### End of the match



#### Super Over

The whole gameplay program is same, except for the following few changes:

* There is no toss. If the player batted first in the tied match, he/she will field first in the super over and vice versa.
* There is only one over per innings, as the name suggests.
* Each team has a maximum of two wickets.
* As per revision to the ICC rules following the 2019 World Cup Final which England won on boundary count, super over will be repeated if it is also tied, until there is a winner.
* In this super over component, the number of games played won’t change. There are also measures to protect this component from being misused.

# Gameplay instructions

## Setting up a team

* Open the python file ‘setupateam.py’.
* Enter a team name of your choice. Before you proceed, please ensure that your team name is not present in the same folder. For example, if a team name ‘2’ is present, the target file is ‘team2.txt’. Therefore, if you want to name your team as ‘2’, the file ‘team2.txt’ should not be there in the same folder.
* Give names to your team members. The names need not be all distinct. However, for convenience, try to avoid names starting with ‘CPU’ followed by a number from 1 to 11, both included.
* A passcode (team key) will be generated. Keep the passcode safe. For safety, carry your team’s text file with you, i.e. create a backup of it. Suppose your team name is ‘0’, create a backup of ‘team0.txt’ file. In case you forget the password (team key), open the text file and check the last string. To avoid password theft, remove your team’s text file from the current folder and store it elsewhere, preferably in an external storage device or online.
* In the example below, the team’s text file is , the Player ID represents the name assigned to each player, while the team’s password is . Once you see something like this (the password and team details may differ), hit ‘Enter’ key or its equivalent(s) to complete the process.

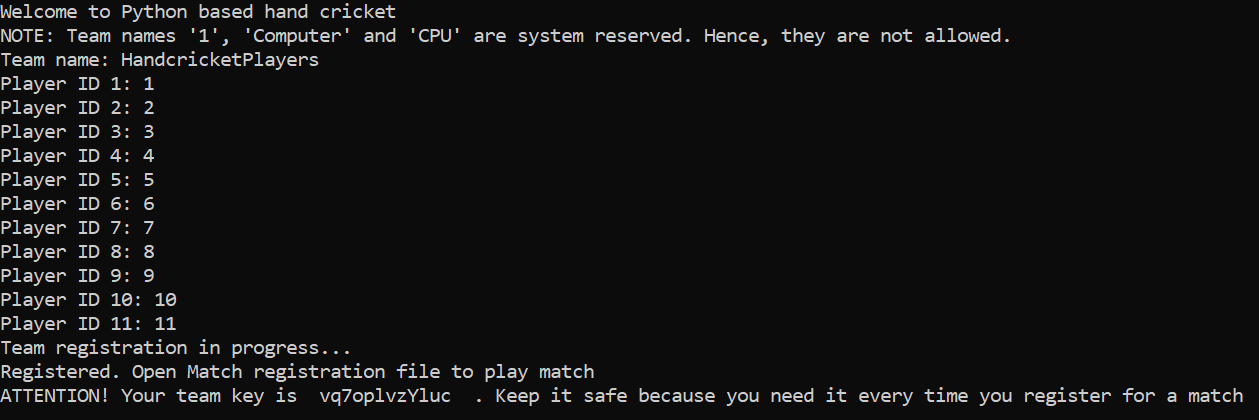


Figure 1: Register team with software: Successful registration message

## Registering for a match

* Once your team is set, you should register for a match before you can play. This is because several teams may exist, while each team would want its personalized experience.
* For this, open ‘registerformatch.py’.
* This is very simple: Enter your team’s name and your team key.

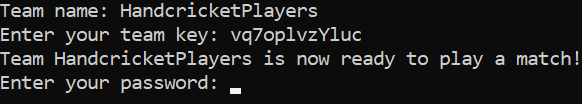


Figure 2: Register a team for a match against the computer: Successful registration message: Choose a password and enter it to proceed.

* If you see an option to enter the password (OTP), enter a password of your choice and hit ‘Enter’ key or its equivalent(s) to complete the process.

## Gameplay

* Now, you are ready to play! Open ‘handcricketgame.py’ to start playing. Your opponent is computer controlled.
* First, you should enter the OTP which you entered while registering for the match.
* Next, you’ll get to choose the number of overs for the match. The default is a 20 over match. Your input should be a positive integer. Otherwise (or if you hit ‘Enter’ without giving any input), the game will last for 20 overs. There is no test match.



Figure 3: Password verified. In this example, the password chosen was ‘password1'. Now choose the number of overs of the game.

* Now choose the number of wickets for the match. It must be a positive integer from 1 to 10, both included. Otherwise (or if you hit ‘Enter’ without giving any input), the game will last for 10 wickets.

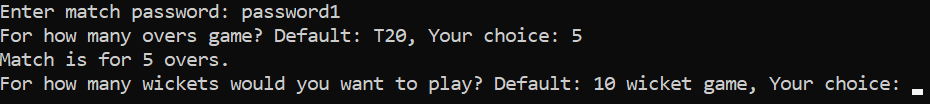


Figure 4: Decided number of overs. For example, this match has 5 overs per innings. Now choose the number of wickets.

* Now it’s time for the toss. Choose a number from 0 to 6. Then choose ‘Odd’ or ‘Even’. Write exactly the same words in the same uppercase/lowercase order. In case of invalid input, you’ll automatically lose the toss.

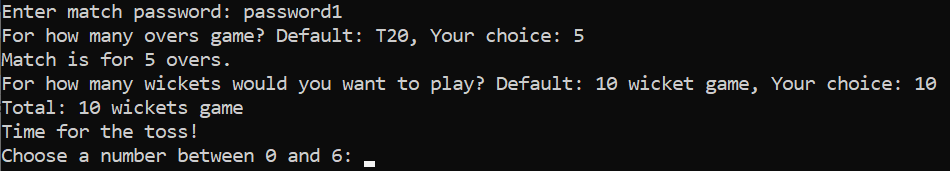


Figure 5: Time for the toss! Follow the instructions. You can choose 0, 1, 2, 3, 4, 5 or 6 as your input.

* If you win the toss, you can choose whether to bat first or field first. Choose wisely. Again, write exactly the same words in the lowercase order only. This time, don’t attempt any experiments, for the application will crash in case of invalid input.

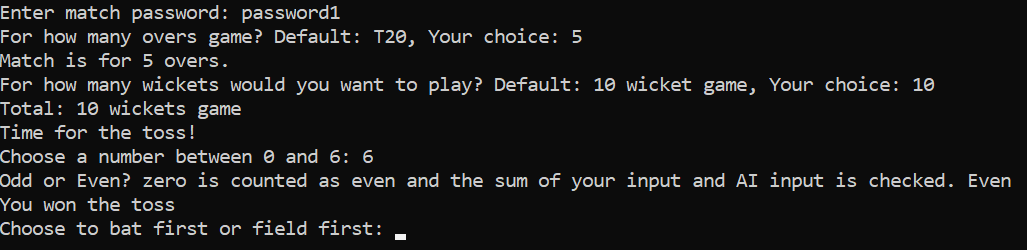


Figure 6: Won the toss. Write the exact word 'bat' or 'field'. If you want to bowl first, input 'field'.

CAUTION: Write everything in only lowercase, otherwise the game won’t run properly.

* If you’re batting, first choose your batters from the list of available batters. Then, just input the number of runs that you want to score and that will add to your score. But here’s the catch: The number must be an integer between 0 and 6, both included. Your opponent must also input any integer in the same range. Thus, if your number matches with your opponent’s number, you are out. Your opponent’s number is hidden, so choose wisely. Don’t worry, invalid input results in no run.

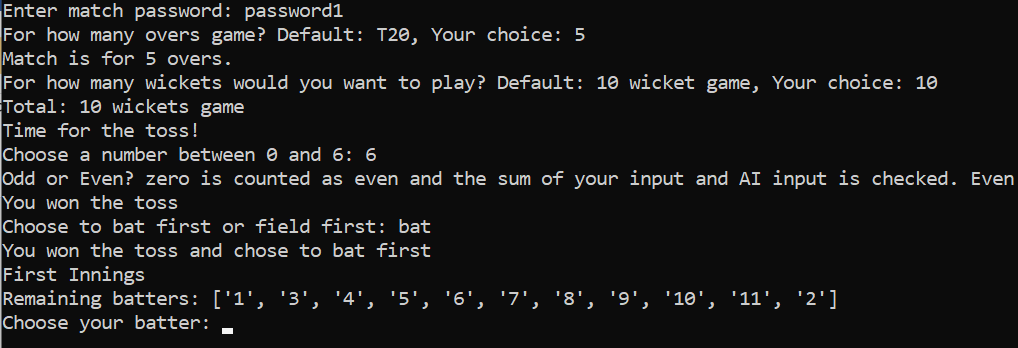


Figure 7: Batting first. Choose your batter

* Note that if your choice of batter/bowler is invalid, a random player from your available list will be selected.

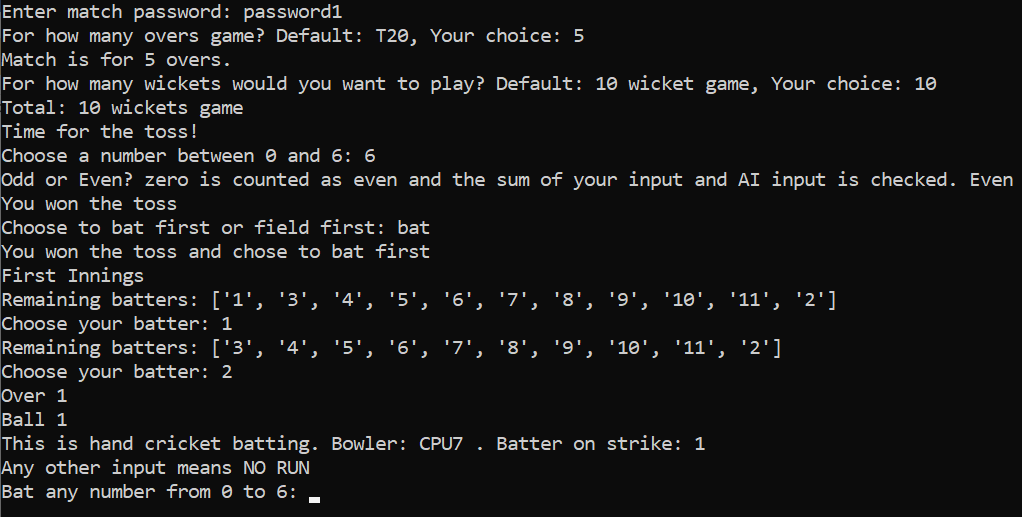


Figure 8: First ball!

* At the end of each over, hit ‘Enter’ or its equivalent(s) to proceed. You can see team score only after any over. At the end of each innings, the innings summary will be displayed.

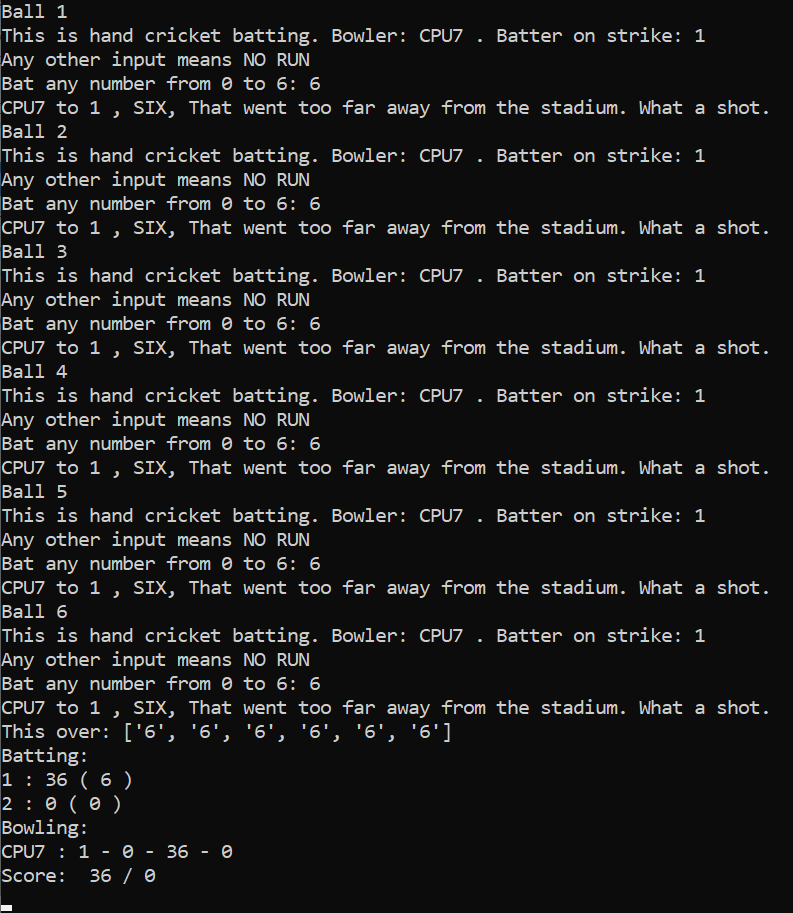


Figure 9: Six sixes in the first over, 36 without loss. Note that the commentary is the same for all the sixes. The commentary is different for different scores and different for every dismissal. Now, hit the ‘Enter’ button to begin the next over. The batters interchange. More action coming up…

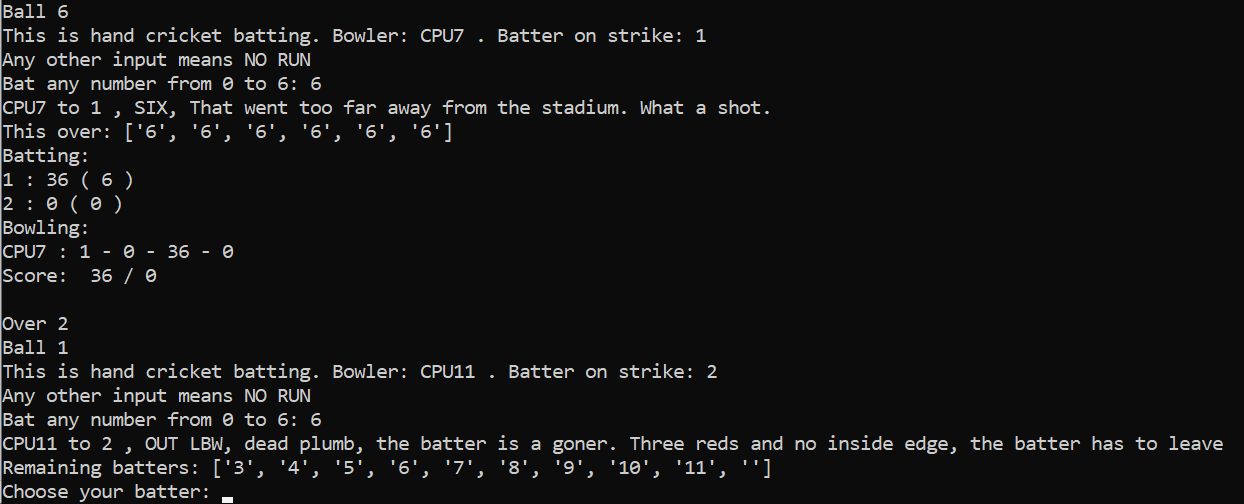


Figure 10: OUT!! LBW and gone for a golden duck! That's why you may not hit 6 all the time!!! Notice the ‘’ with the remaining batters. If it was omitted, the program would crash if all 10 wickets fell. Now it won’t, since we have the blank. Just be sure to manually choose the batter so that the blank does not bat. Don’t ever choose the blank!

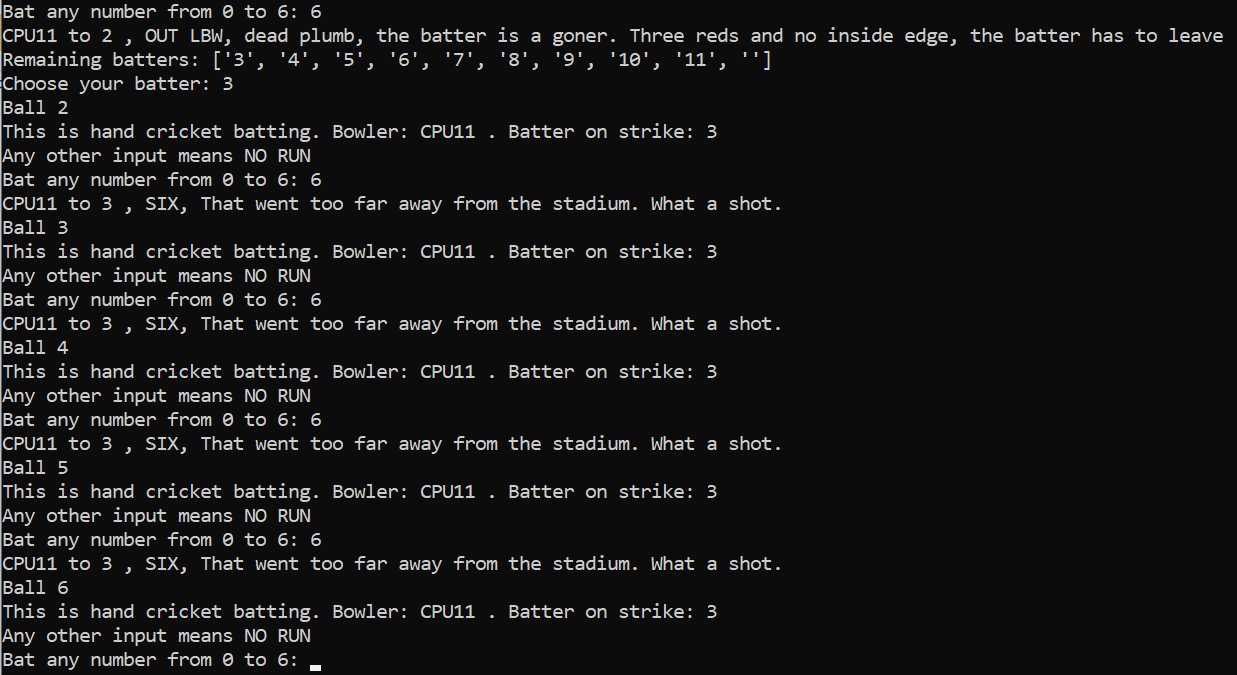


Figure 11: Still going for sixes? Batter number 3 is taking a huge risk! Nevertheless, there are only 5 overs per side, so it may be worth the risk.

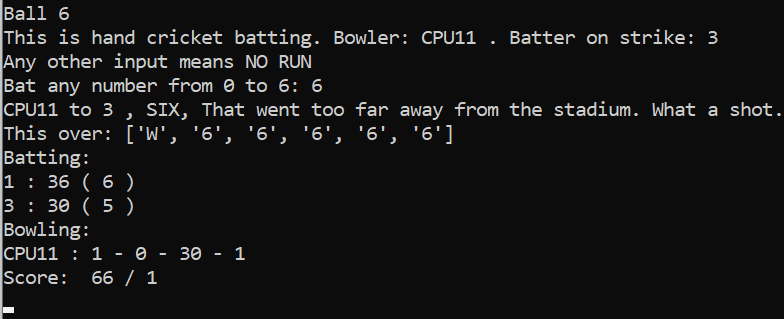


Figure 12: This is what the bowler's figures would look like.

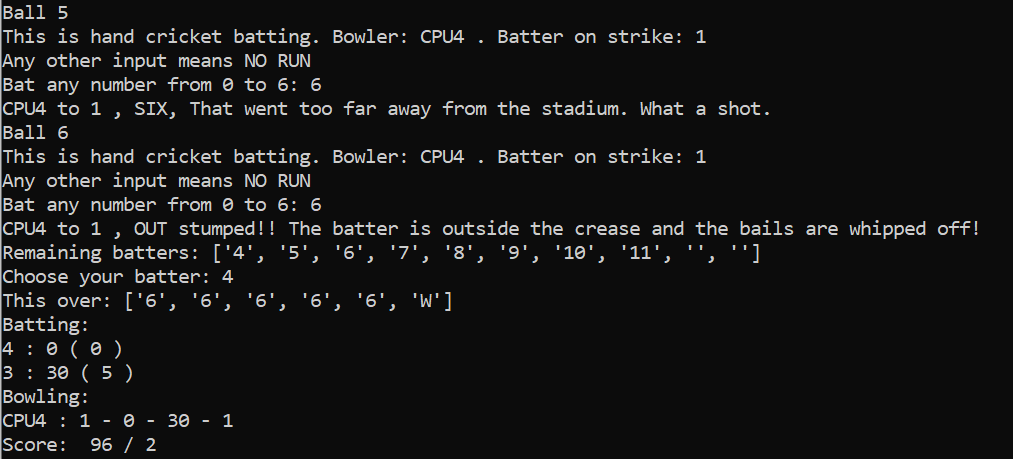


Figure 13: Last ball of the third over, finally gone! Batter 1 has hit 66 runs off 12 deliveries

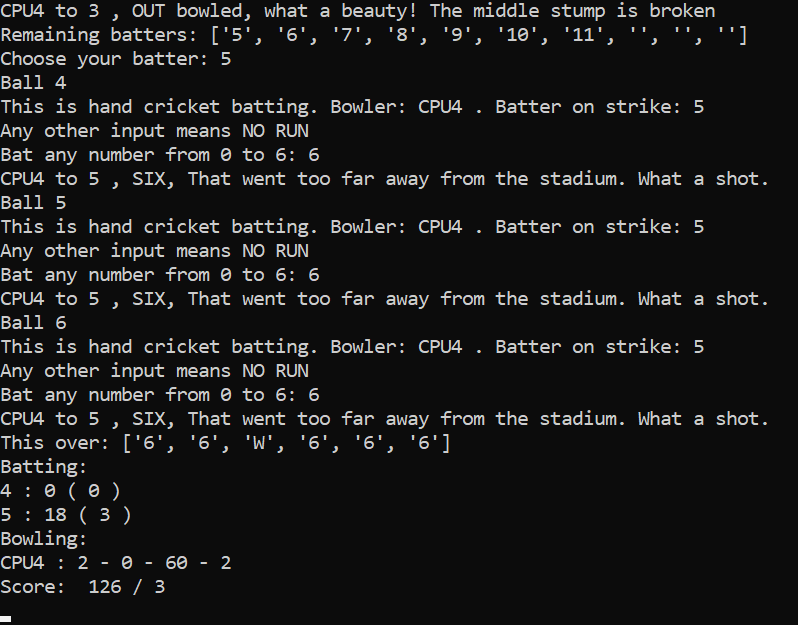


Figure 14: Penultimate over: one batter out, a bowler claims two wickets. Only one over remains...

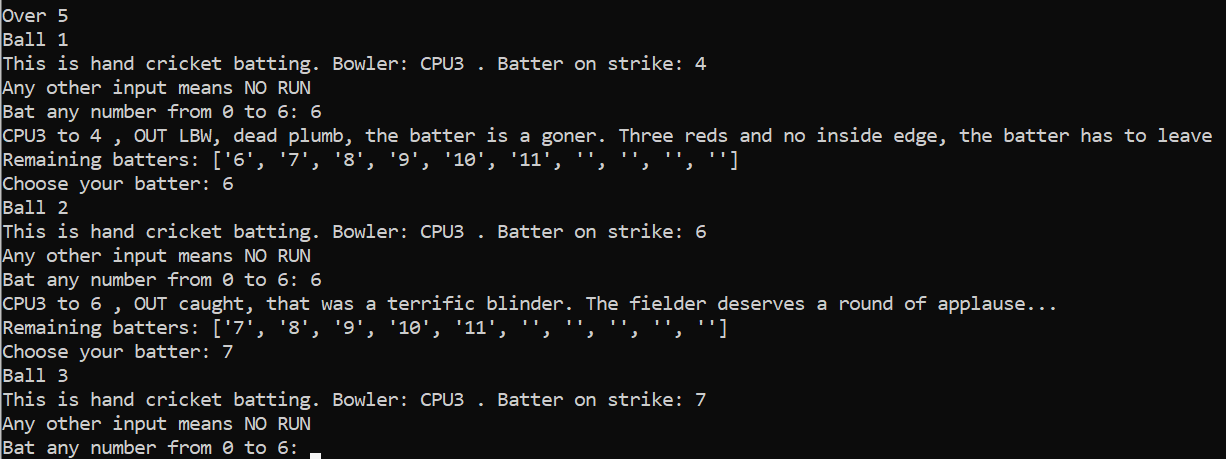


Figure 15: Two wickets in two balls! First LBW, then a stunning catch, the bowler is on a hat-trick. Will he claim it?

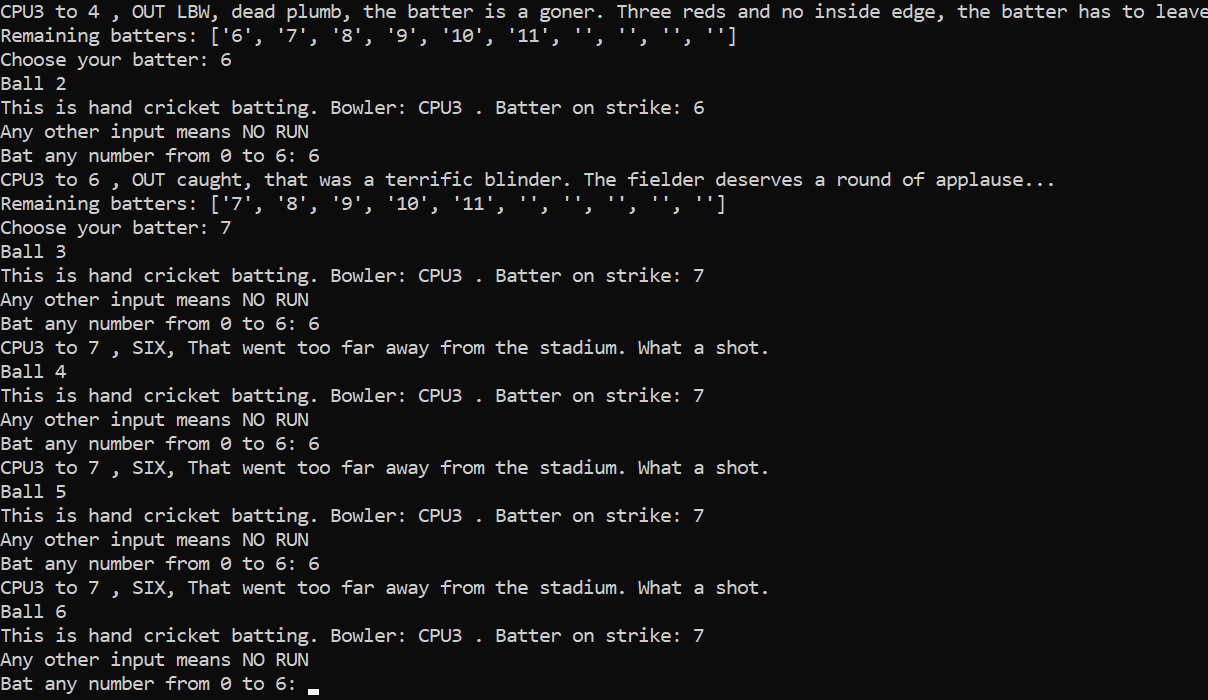


Figure 16: No hat-trick! The last ball of this innings...

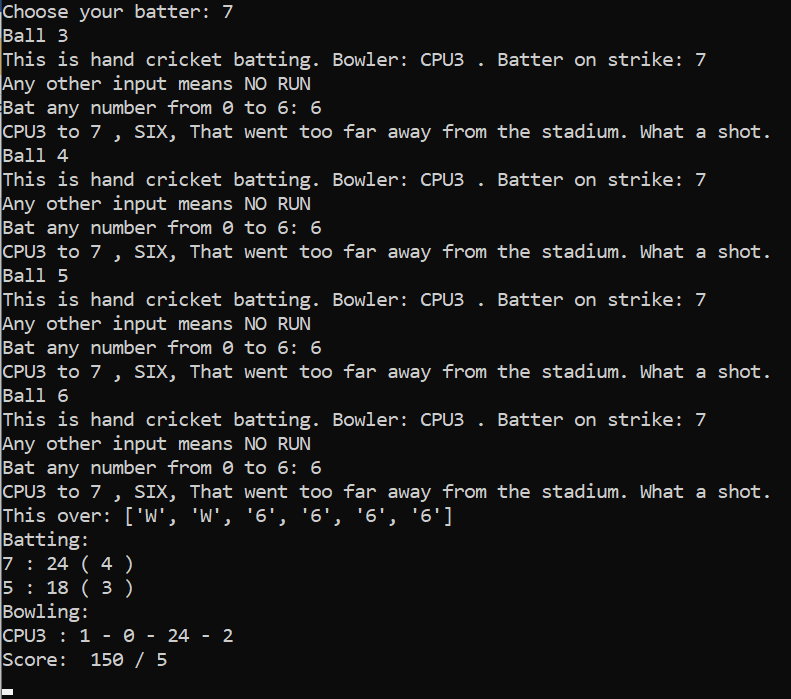


Figure 17: End of the first innings! We finish at 150-5

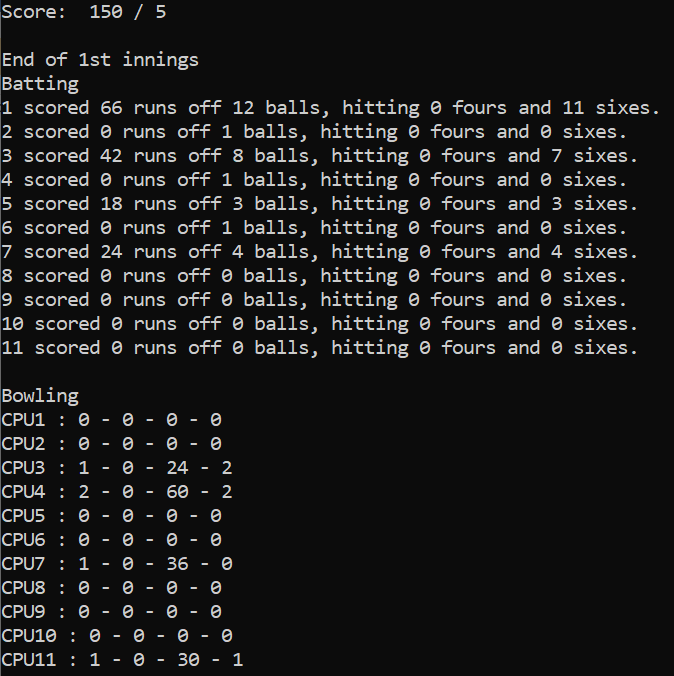


Figure 18: First innings summary

* Similarly, if you’re bowling (fielding), choose a bowler and input a number in the same range, but you would want your number to match with your opponent’s number to get him out. This time if you give any other integer, your opponent will get a six. If you give any other invalid input, your opponent will score as many runs as its input without getting out. Note that a bowler cannot bowl two consecutive overs and a bowler cannot bowl more than one-fifth of the total number of match overs. For example, in a 20 over game, no bowler is permitted more than 4 overs.

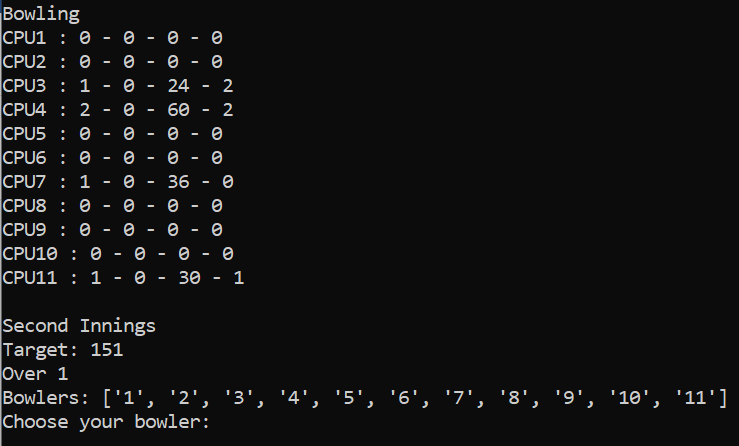


Figure 19: Choose your bowler. The target in this example is 151 runs.

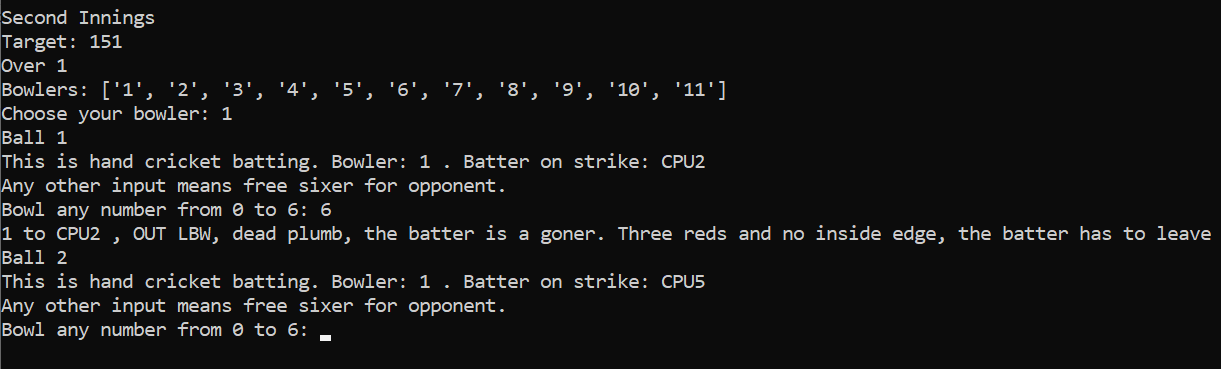


Figure 20: There we go! We claim a wicket in the very first ball of the second innings!

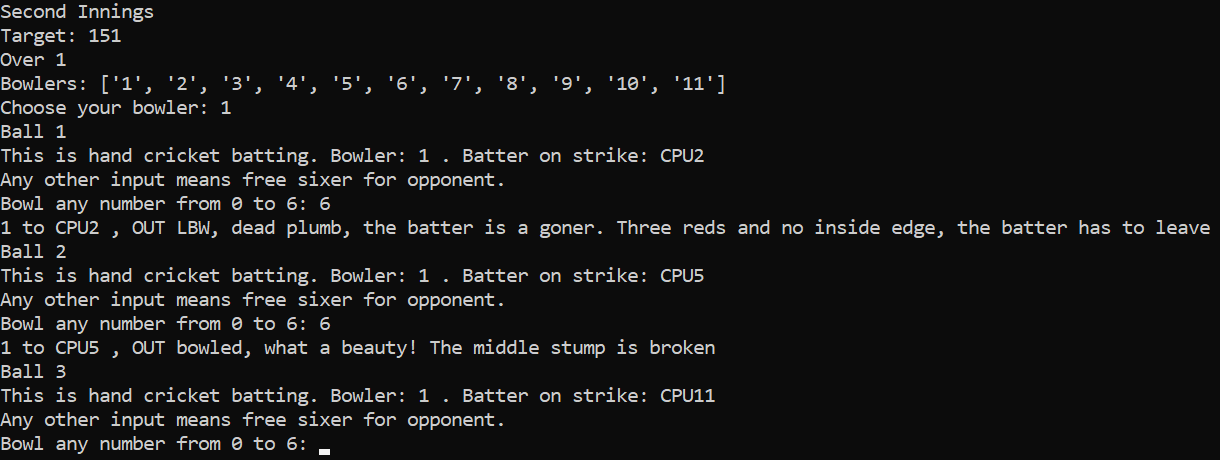


Figure 21: Again, a wicket! Now, we're on a hat-trick! Will we get it?



Figure 22: No hat trick, we're out of luck. We expected our opponent to hit six all the time. This time, he responded with 3. Very clever indeed!

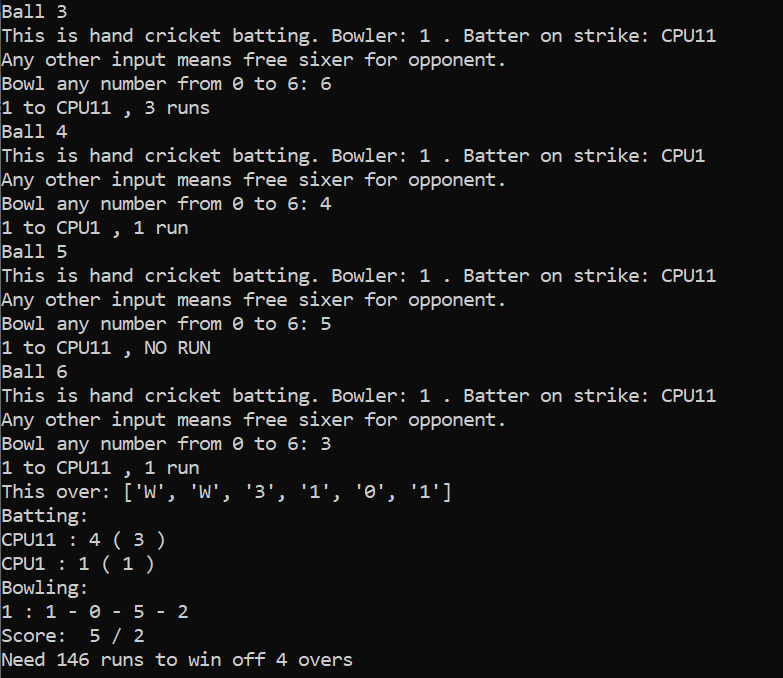


Figure 23: Responding with different inputs. We have a theoretically won game now, since his required run rate is more than 36 runs per over.

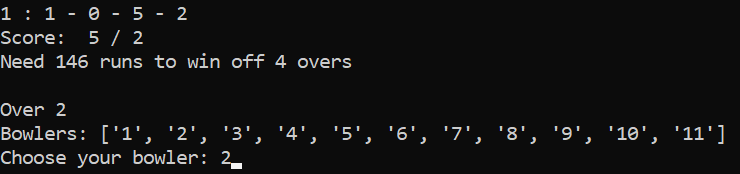


Figure 24: Second over. This list is the list of all bowlers. We can't choose bowler 1 now. Let's pick 2 and see what happens...

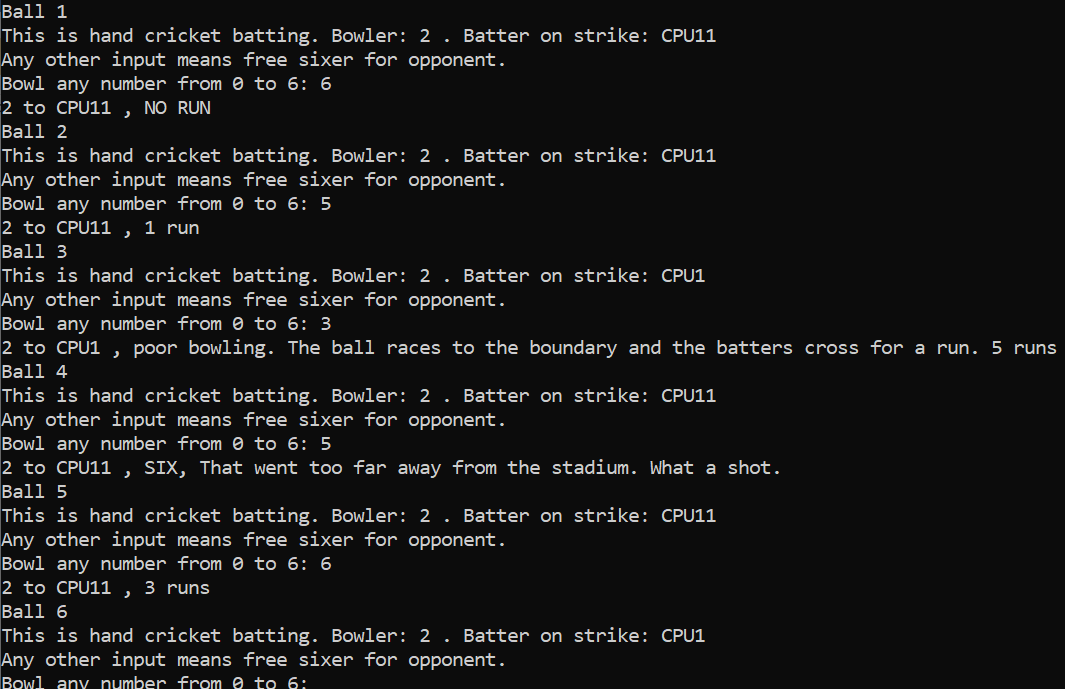


Figure 25: Our bowler in action...

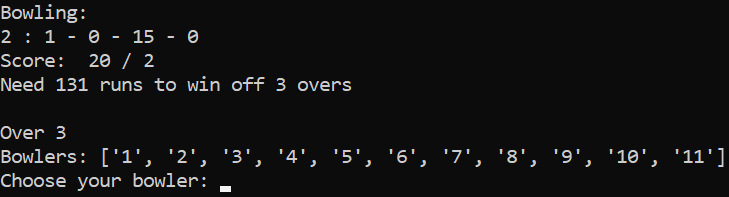


Figure 26: Can't pick bowler 2. Bowler 1 is free, but each bowler has only one over since there are only 5 overs! Let's pick bowler 3.

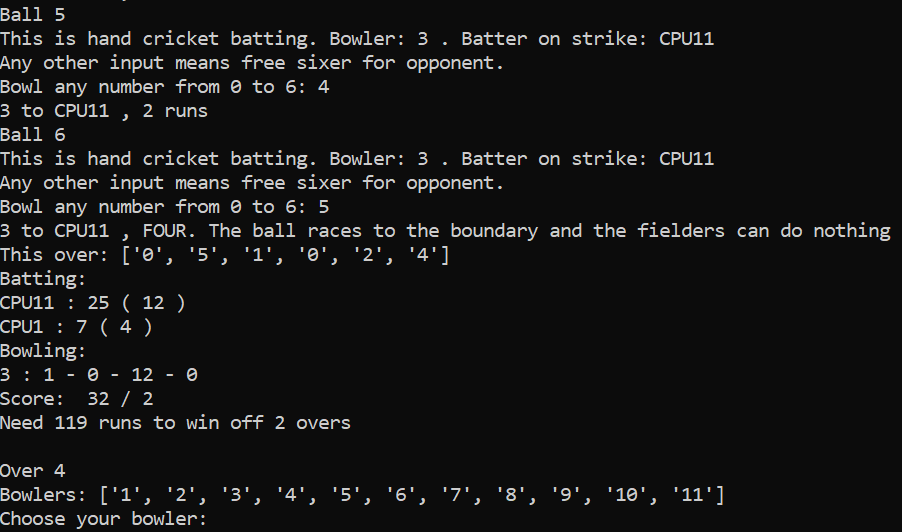


Figure 27: Now let's have fun!!! Bowler 4 will bowl the next over for us.

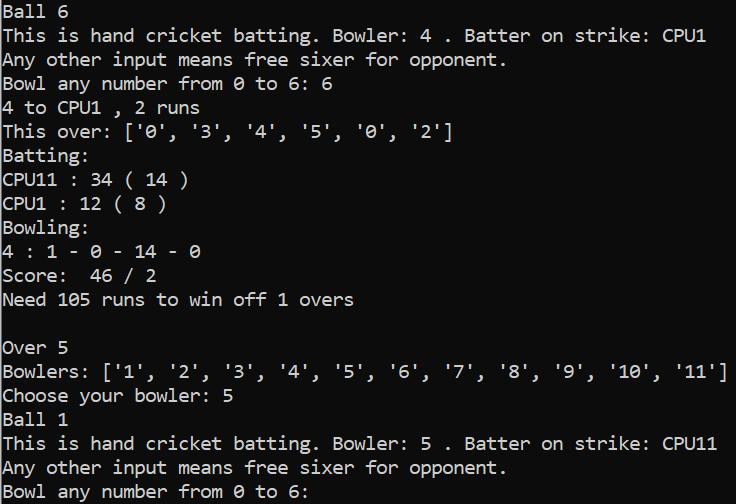


Figure 28: Last over. 105 to win. Impossible! Now it's time to perform the final rites of this match.

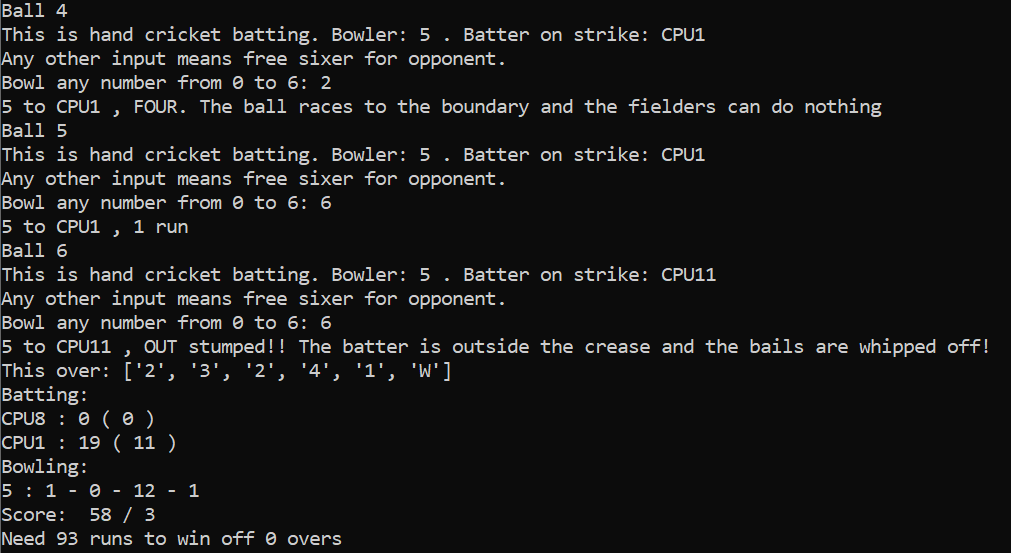


Figure 29: That's it! We win the match.

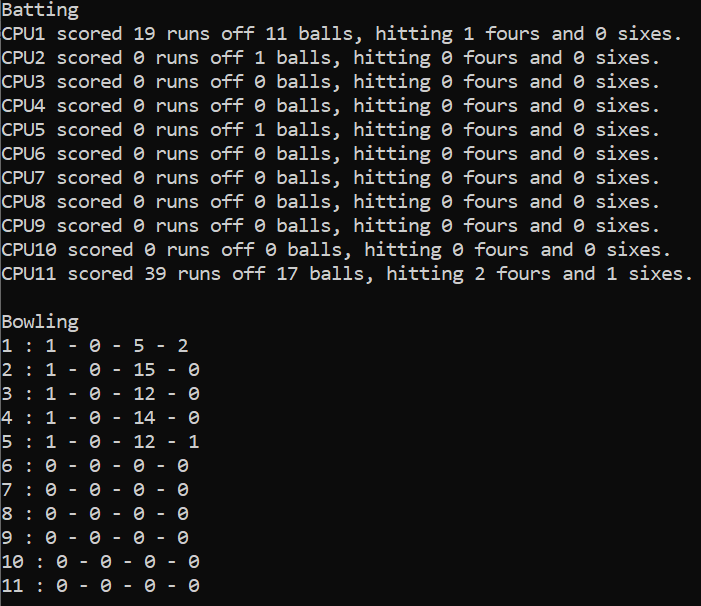


Figure 30: The opponent's batting. We don't have to explicitly declare end of second innings. We win by 92 runs

* Score more than your opponent to win. At the end of the match, the total number of matches that you played and the total number of successful victories will be displayed.

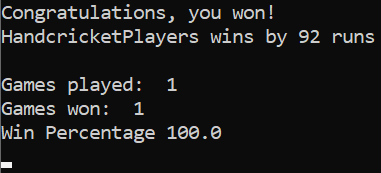


Figure 31: Won the match. This is the winning screenshot.

* In the event of a tie, you have the option of playing super over. Another passcode called the super over key is generated. You will have to keep both match password and super over key safe as you require it to play the super over. If you start another match straightaway without completing the super over, you will be considered to have resigned the match and hence lost. The super over key can be used only once.

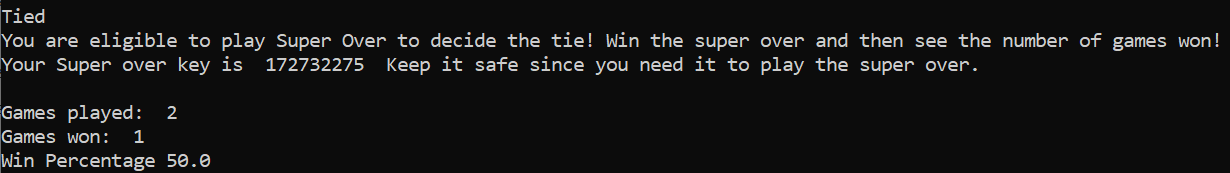


Figure 32: This is what happens if the scores are same. We batted first.

# Super Over

* The super over is the official way to settle ties. The gameplay for super over is as follows:

## Gameplay

* If you have tied your match, open ‘handcricketgamesuperover.py’ to start playing. You will face the same opponent.
* First, you should enter the OTP which you entered while registering for the tied match as well as the super over key.

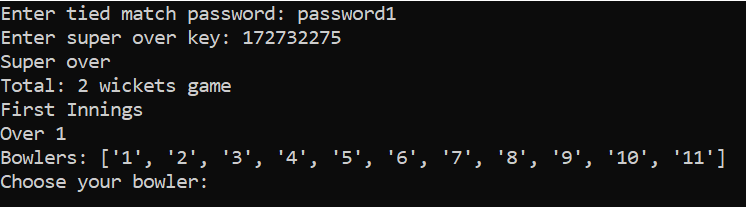


Figure 33: Starting the super over. There is no toss. Since we batted first in the tied match, we bowl first.

* Each team gets only 1 over and 2 wickets
* There is no toss. If you batted first in the tied match, you will field first in this super over. If you batted second in the tied match, you will bat first in this super over.
* If you’re batting, first choose your batters from the list of available batters. Then, just input the number of runs that you want to score and that will add to your score. But here’s the catch: The number must be an integer between 0 and 6, both included. Your opponent must also input any integer in the same range. Thus, if your number matches with your opponent’s number, you are out. Your opponent’s number is hidden, so choose wisely. Don’t worry, invalid input results in no run.
* Similarly, if you’re bowling (fielding), choose a bowler and input a number in the same range, but you would want your number to match with your opponent’s number to get him out. This time if you give any other integer, your opponent will get a six. If you give any other invalid input, your opponent will score as many runs as its input without getting out. Note that only one bowler is permitted throughout the innings.
* Note that if your choice of batter/bowler is invalid, a random player from your available list will be selected.
* At the end of each over, hit ‘Enter’ or its equivalent(s) to proceed. You can see team score only after any over. At the end of each innings, the innings summary will be displayed.
* Score more than your opponent to win. At the end of the match, the total number of matches that you played and the total number of successful victories will be displayed.
* If you win, the number of games won will be incremented accordingly.
* There is no change to the number of games played, since the super over is only a tiebreaker.

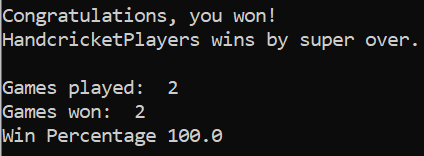


Figure 34: Winning the super over. The tie is broken.

* In the event of a tie, you have the option of playing super over again. Another super over key is generated. You will have to keep both match password and super over key safe as you require it to play the super over. If you start another match straightaway without completing the super over, you will be considered to have resigned the match and hence lost. The super over key can be used only once.

# References:

<https://www.instructables.com/id/How-to-Play-Hand-Cricket/>

<https://www.python.org/downloads/release/python-374/>

# Original Project Developer:

This Hand Cricket Project was originally developed by Burra Abhishek.

Contact: +91 9884043460

E-Mail: [burra.abhishek1@gmail.com](mailto:burra.abhishek1@gmail.com)