# **Problem/Task :-**

**Blind Auction**

Click "Open Preview" above to see this file rendered with the markdown.

**Instructions**

The objective is to write a program that will collect the names and bids of different people. The program should ask for each bidder's name and their bid individually.

Welcome to the secret auction program.

What is your name?: Angela

What's your bid?: $123

Are there any other bidders? Type 'yes' or 'no'.

yes

If there are other bidders, the screen should clear, so you can pass your phone to the next person. If there are no more bidders, then the program should display the name of the winner and their winning bid.

The winner is Elon with a bid of $55000000000

Use your knowledge of Python dictionaries and loops to solve this challenge.

**My console doesn't clear!**

This will happen if you’re using an IDE other than replit (e.g., VSCode, PyCharm etc). Similar to how we used the "random" module previously, in this project we will use the "replit" module. The clear() function is available here via the replit module without any extra configuration.

**I’ll cover how to use PyCharm and import modules on Day 15**. That said, you can write your own clear() function or configure your IDE like so:

[Udemy Q&A Answer](https://www.udemy.com/course/100-days-of-code/learn/lecture/19279420#questions/16084076)

**Solution**

[https://replit.com/@appbrewery/blind-auction-completed](https://replit.com/@appbrewery/blind-auction-completed?v=1)

A blue background with white text

Description automatically generated with low confidence



# **Flow Chart:**

|  |
| --- |
|  |

# **Code:**

**# from replit import clear**

**#HINT: You can call clear() to clear the output in the console.**

**import os**

**from Art import logo**

**print(logo)**

**def check(check\_other\_bidders):**

**if check\_other\_bidders.lower() == 'yes':**

**return True**

**else:**

**return False**

**def bid\_record():**

**bidding\_record = {}  # Empty dictionary to store the bidders and their bids**

**check\_other\_bidders = input("Type 'yes' if there are other bidders, or 'no' to finish: ")**

**while check(check\_other\_bidders):**

**bidder = input("What is your Name?: ")**

**bid\_amount = float(input("What is your bid?: $"))**

**bidding\_record[bidder] = bid\_amount  # Add the bidder and their bid to the record**

**check\_other\_bidders = input("Type 'yes' if there are other bidders, or 'no' to finish: ")**

**os.system('clear')  # Clears the screen before taking the second bid**

**# print("Bidding record:", bidding\_record)**

**highest\_bidder = ''**

**highest\_bid = 0**

**# Find the highest bidder and their bid in the record**

**for bidder, bid in bidding\_record.items():**

**if bid > highest\_bid:**

**highest\_bidder = bidder**

**highest\_bid = bid**

**print("Highest bidder:", highest\_bidder)**

**print("Highest bid:", highest\_bid)**

**bid\_record()**

This code is written in Python and it creates a function called bid\_record(). The purpose of this function is to create a bidding record that stores the bidders and their bids, and then find the highest bidder and their bid from the record.

At the beginning of the function, an empty dictionary called bidding\_record is created to store the bidders and their bids. The function then prompts the user to enter whether there are other bidders or not. The response is stored in a variable called check\_other\_bidders. This is done using the input() function in Python.

The while loop is used to keep prompting the user to enter their name and bid amount until there are no more bidders. The check() function is called in the while loop as a condition to check whether there are more bidders or not. It is not defined in the code snippet provided, so it is assumed that it is defined elsewhere in the code.

Inside the while loop, the user is prompted to enter their name and bid amount using the input() function. The float() function is used to convert the bid amount to a float data type. The bidder's name and bid amount are then added to the bidding\_record dictionary using the bidding\_record[bidder] = bid\_amount syntax.

After each bidder makes their bid, the user is prompted again to enter whether there are more bidders or not. The os.system('clear') function is used to clear the screen before taking the second bid. This function is used to provide a cleaner interface for the user.

Once there are no more bidders, the for loop is used to find the highest bidder and their bid in the bidding\_record dictionary. The items() method is used to iterate over the dictionary and return both the key-value pairs. The if statement checks whether the current bid is greater than the previous highest bid. If it is, the current bidder becomes the highest bidder and their bid becomes the new highest bid.

Finally, the print() function is used to print the highest bidder's name and their bid to the console.

# **My Replit:-**

[main.py - blind-auction-start - Replit](https://replit.com/@BiswapradipDas/blind-auction-start#main.py)

# **Python Tutor:-**

[Python Tutor code visualizer: Visualize code in Python, JavaScript, C, C++, and Java](https://pythontutor.com/visualize.html#mode=edit)