# Movie budgee

# Full Source Code

movies = [  
 ("Eternal Sunshine of the Spotless Mind", 20000000),  
 ("Memento", 9000000),  
 ("Requiem for a Dream", 4500000),  
 ("Pirates of the Caribbean: On Stranger Tides", 379000000),  
 ("Avengers: Age of Ultron", 365000000),  
 ("Avengers: Endgame", 356000000),  
 ("Incredibles 2", 200000000)  
]  
  
want = input("you want to add or not")  
if want =="yes":  
 num = int(input("how many movie you want to add"))  
 for i in range(num):  
 movie = input("enter movie name")  
 budget = int(input("enter budget"))  
 movies.append((movie,budget))  
elif want == "no":  
 print("as you wish")  
 exit  
  
  
total\_budget = 0  
for movie,budget in movies:   
 total\_budget = total\_budget + budget  
 average = total\_budget / len(movies)  
print("total expense is :" ,total\_budget)   
print("average is :",average)  
  
  
for movie,budget in movies:   
 if budget > average:  
 print(f"movie {movie} and budget greater than average is {budget}")  
   
count = 0  
for movie,budget in movies:   
 if budget > average:   
 print(f"Movie {movie} has budget {budget}, which is {budget - average} higher than the average {average}")  
 count += 1  
   
   
print(f"number of movie above average is : {count}")

# How the Code Works

1. The program starts with a predefined list called `movies`. Each item is a tuple: (movie\_name, budget).

2. It asks the user: "you want to add or not". If the user types "yes", it asks how many movies to add.

3. When adding movies, for each one it asks for the title (`movie`) and the `budget`, then appends `(movie, budget)` to the list.

4. If the user types "no", it prints "as you wish" and then intends to exit (note: this should be `exit()` to actually stop).

5. It initializes `total\_budget = 0`.

6. It loops through all `(movie, budget)` pairs in `movies` and keeps adding each `budget` to `total\_budget`.

7. Inside the same loop, it computes `average = total\_budget / len(movies)`. (This keeps recomputing the average each iteration; the final value after the loop finishes is the overall average.)

8. It prints the final `total\_budget` and the `average`.

9. Next, it loops through the movies again and prints those where `budget > average` (i.e., budgets above the average).

10. It sets `count = 0` and does another loop through the movies.

11. For each movie with `budget > average`, it prints how much higher the budget is compared to the average and increments `count`.

12. Finally, it prints how many movies were above the average using `count`.