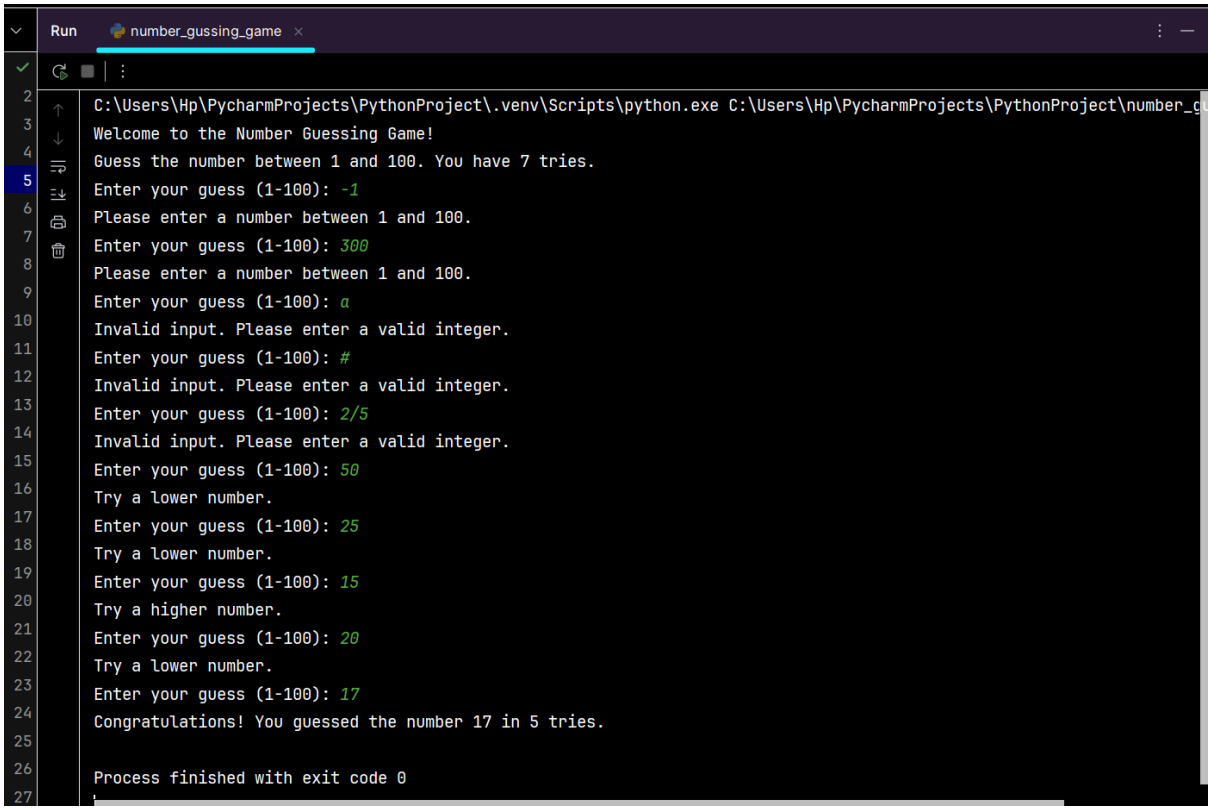
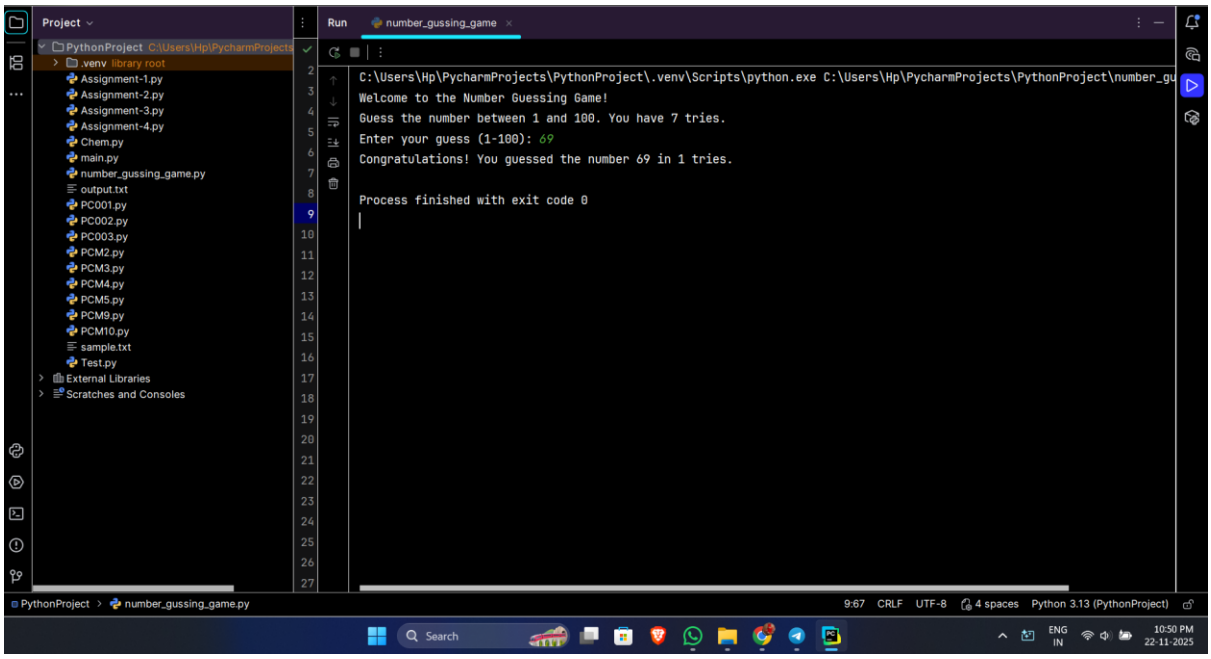
Screen Shots :-

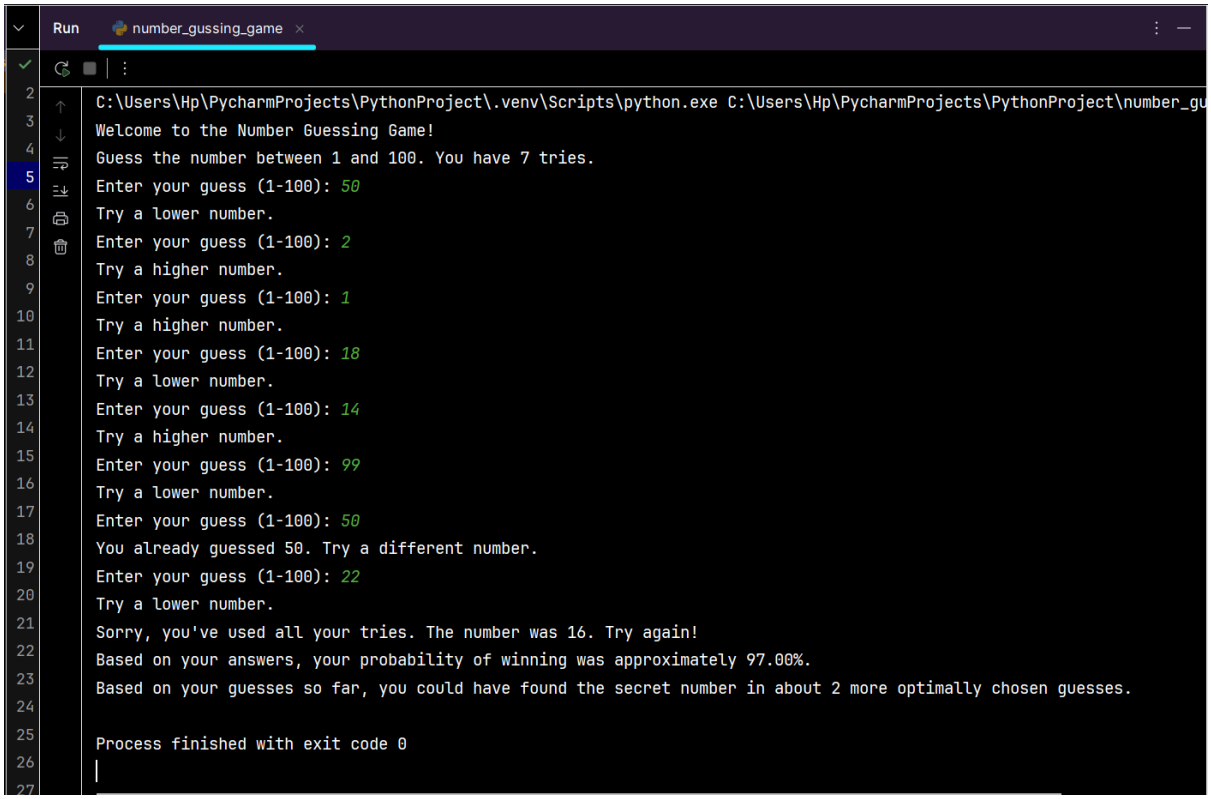
```python
import random
import sys
import time


def get_valid_guess(lower, upper, previous_guesses):  1 usage
    while True:
        try:
            guess = input(f"Enter your guess ({lower}-{upper}): ")
            guess = int(guess)
            if guess in previous_guesses:
                print(f"You already guessed {guess}. Try a different number.")
                time.sleep(1)
                continue
            if guess < lower or guess > upper:
                print(f"Please enter a number between {lower} and {upper}.")
                time.sleep(1)
                continue
            return guess
        except ValueError:
            print("Invalid input. Please enter a valid integer.")
            time.sleep(1)


def analyze_guesses(guesses, secret, lower, upper):  1 usage
    possible = set(range(lower, upper + 1))
    for g in guesses:
        if g < secret:
            possible = set(x for x in possible if x > g)
        elif g > secret:
            possible = set(x for x in possible if x < g)
    remaining = len(possible)
    total = upper - lower + 1
    prob = 1 - remaining / total
    return max(0, min(1, prob))


def min_additional_tries(guesses, secret, lower, upper):  1 usage
    # Calculate how many more tries would be needed assuming an optimal binary search on the reduced range
    possible = set(range(lower, upper + 1))
    for g in guesses:
        if g < secret:
            possible = set(x for x in possible if x > g)
        elif g > secret:
            possible = set(x for x in possible if x < g)
    remaining = len(possible)
    # Number of tries in worst-case binary search = ceil(log2(remaining))
    import math
    if remaining <= 1:
        return 0
    return math.ceil(math.log2(remaining))

```

```
51      return math.ceil(math.log2(remaining))
52
53
54  def number_guessing_game():  1 usage
55      lower = 1
56      upper = 100
57      max_attempts = 7
58      secret = random.randint(lower, upper)
59      guesses = []
60
61      print(f"Welcome to the Number Guessing Game!")
62      time.sleep(1)
63      print(f"Guess the number between {lower} and {upper}. You have {max_attempts} tries.")
64
65      for attempt in range(1, max_attempts + 1):
66          guess = get_valid_guess(lower, upper, guesses)
67          guesses.append(guess)
68
69          if guess < secret:
70              print("Try a higher number.")
71          elif guess > secret:
72              print("Try a lower number.")
73          else:
74              print(f"Congratulations! You guessed the number {secret} in {attempt} tries.")
75              sys.exit(0)
76          time.sleep(1)
77      else:
```

```
54  def number_guessing_game():  1 usage
71          elif guess > secret:
72              print("Try a lower number.")
73          else:
74              print(f"Congratulations! You guessed the number {secret} in {attempt} tries.")
75              sys.exit(0)
76          time.sleep(1)
77      else:
78          print(f"Sorry, you've used all your tries. The number was {secret}. Try again!")
79
80      probability = analyze_guesses(guesses, secret, lower, upper)
81      print(f"Based on your answers, your probability of winning was approximately {probability:.2%}.")
82
83      more_tries_needed = min_additional_tries(guesses, secret, lower, upper)
84      print(f"Based on your guesses so far, "
85            f"you could have found the secret number in about {more_tries_needed} more "
86            f"optimally chosen guess{'es' if more_tries_needed != 1 else ''}.")
87
88
89  if __name__ == "__main__":
90      number_guessing_game()
91
```


Run
number_gussing_game
C:\Users\Hp\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\Hp\PycharmProjects\PythonProject\number_g
Welcome to the Number Guessing Game!
Guess the number between 1 and 100. You have 7 tries.
Enter your guess (1-100): -1
Please enter a number between 1 and 100.
Enter your guess (1-100): 300
Please enter a number between 1 and 100.
Enter your guess (1-100): a
Invalid input. Please enter a valid integer.
Enter your guess (1-100): #
Invalid input. Please enter a valid integer.
Enter your guess (1-100): 2/5
Invalid input. Please enter a valid integer.
Enter your guess (1-100): 50
Try a lower number.
Enter your guess (1-100): 25
Try a lower number.
Enter your guess (1-100): 15
Try a higher number.
Enter your guess (1-100): 20
Try a lower number.
Enter your guess (1-100): 17
Congratulations! You guessed the number 17 in 5 tries.
Process finished with exit code 0


Project
PythonProject C:\Users\Hp\PycharmProjects
.venv library root
Assignment-1.py
Assignment-2.py
Assignment-3.py
Assignment-4.py
Chem.py
main.py
number_gussing_game.py
output.txt
PC001.py
PC002.py
PC003.py
PCM2.py
PCM3.py
PCM4.py
PCM5.py
PCM9.py
PCM10.py
sample.txt
Test.py
External Libraries
Scratches and Consoles
Run
number_gussing_game
C:\Users\Hp\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\Hp\PycharmProjects\PythonProject\number_gu
Welcome to the Number Guessing Game!
Guess the number between 1 and 100. You have 7 tries.
Enter your guess (1-100): 69
Congratulations! You guessed the number 69 in 1 tries.
Process finished with exit code 0
PythonProject
number_gussing_game.py
9:67
CRLF
UTF-8
4 spaces
Python 3.13 (PythonProject)
Search
ENG
IN
10:50 PM
22-11-2025


Run
number_gussing_game
C:\Users\Hp\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\Hp\PycharmProjects\PythonProject\number_gu
Welcome to the Number Guessing Game!
Guess the number between 1 and 100. You have 7 tries.
Enter your guess (1-100): 50
Try a lower number.
Enter your guess (1-100): 2
Try a higher number.
Enter your guess (1-100): 1
Try a higher number.
Enter your guess (1-100): 18
Try a lower number.
Enter your guess (1-100): 14
Try a higher number.
Enter your guess (1-100): 99
Try a lower number.
Enter your guess (1-100): 50
You already guessed 50. Try a different number.
Enter your guess (1-100): 22
Try a lower number.
Sorry, you've used all your tries. The number was 16. Try again!
Based on your answers, your probability of winning was approximately 97.00%.
Based on your guesses so far, you could have found the secret number in about 2 more optimally chosen guesses.
Process finished with exit code 0