



**Project Title : Cows And Bulls**

**Name: Devanshu Singh**

**Roll No: 2210990263**

- Objective
- Introduction
- Methodology, Approach & Techniques
- Algorithm
- Result
- Source Code (screenshots)
- Conclusion
- Reference

- **Objective of the Bulls and Cows Game**
- This game is a number guessing game where the player tries to crack a secret 4-digit number within a set number of attempts.
- **Secret Number:** The game generates a random 4-digit number where each digit is unique (no repetitions).
- **Player's Guess:** In each attempt, the player needs to guess a 4-digit number with unique digits.

- **Bulls and Cows: Crack the Code!**
- Are you ready to test your logic and deduction skills? We're about to play a mind-bending game called Bulls and Cows!
- Imagine a secret 4-digit number, just waiting to be cracked. Each digit in this number is unique, there are no sneaky repeats! Your mission, should you choose to accept it, is to guess this secret number within a set number of tries.
- But don't worry, you won't be left completely in the dark. After each guess, you'll receive helpful clues in the form of bulls and cows. These clues will guide you closer to the hidden code, with:
- **Bulls:** Representing the digits that are absolutely correct, both the number itself and its position in the secret code.
- **Cows:** Highlighting digits that are present in the secret code, but are unfortunately in the wrong spot!
- So, are you ready to unleash your inner detective and crack the code? Let's play Bulls and Cows!

- **Methodology:**
- **Hidden Information:** The core methodology relies on hidden information, where the secret 4-digit number is generated and kept hidden from the player.
- **Approach:**
- **Trial and Error with Feedback:** The game follows a trial-and-error approach with feedback. Players make guesses, receive feedback on their accuracy (bulls and cows), and use this information to refine their subsequent guesses.
- **Techniques:**
- **Random Number Generation:** The `generateNum` function utilizes a random number generation technique to create a secret number that meets the criteria (4-digits, unique digits).
- **Data Validation:** The code employs data validation techniques to ensure player guesses adhere to the game's rules (4-digits and no repeated digits). This is implemented in the `noDuplicates` function and within the guess input loop.
- **Comparison and Counting:** The `numOfBullsCows` function uses comparison and counting techniques to evaluate the player's guess against the secret number. It iterates through corresponding digits in both numbers, identifying bulls (correct digit and position) and cows (correct digit, wrong position).

## Bulls and Cows Algorithm

This algorithm outlines the steps involved in the Bulls and Cows game:

### 1. Secret Number Generation:

- The program generates a random 4-digit number using a function like `generateNum`.
- This function ensures each digit is unique (no repetitions) using techniques like repeatedly generating numbers until this criterion is met.

### 2. Player Input and Validation:

- The program prompts the player to enter the number of tries allowed (e.g., using `input`).
- A loop iterates for the specified number of tries.
  - Inside the loop:
    - The program prompts the player to enter their guess (e.g., using `input`).
    - A validation function like `noDuplicates` checks if the guess is a 4-digit number with no repeated digits.
      - If invalid, the program displays an error message and prompts for a new guess (continue statement skips to the next iteration).
    - If valid, the program proceeds with evaluating the guess.

### 3. Guess Evaluation:

- A function like `numOfBullsCows` compares the player's guess with the secret number.
- It iterates through corresponding digits in both numbers:
  - If a digit in the guess matches a digit in the secret number and they occupy the same position, it's a bull (increment bull counter).
  - If a digit in the guess matches a digit in the secret number, but they are in different positions, it's a cow (increment cow counter).

### 4. Feedback and Win Condition:

- After evaluation, the program displays feedback to the player indicating the number of bulls and cows.
- The program checks if all four digits are bulls (i.e., `bull_cow[0] == 4`).
  - If yes, the player wins, and the program exits the loop with a congratulatory message.

### 5. Loop Continuation or End Game:

- If the player hasn't guessed correctly and there are still tries remaining (as checked by the loop counter), the loop continues to the next iteration, prompting for a new guess.
- If the player runs out of tries without guessing correctly, the program exits the loop and reveals the secret number.

The Bulls and Cows game culminates in a thrilling conclusion, with players either triumphantly cracking the secret code within the allotted tries (displaying a congratulatory message with the number of tries taken) or strategically succumbing to defeat, at which point the game reveals the elusive secret number.

# Source Code (screenshots)



```
File Edit Selection View Go Run ... c++_course
python_project.ipynb project.py x sliding_window.cpp class.ipynb numpy.ipynb
python > project.py > numOfBullsCows
Click here to ask Blackbox to help you code faster |
1 import random
2
3 def getDigits(num):
4     return [int(i) for i in str(num)]
5
6 def noDuplicates(num):
7     num_li = getDigits(num)
8     if len(num_li) == len(set(num_li)):
9         return True
10    else:
11        return False
12
13 def generateNum():
14     while True:
15         num = random.randint(1000,9999)
16         if noDuplicates(num):
17             return num
18
19 def numOfBullsCows(num,guess):
20     bull_cow = [0,0]
21     num_li = getDigits(num)
22     guess_li = getDigits(guess)
23
24     for i,j in zip(num_li,guess_li):
25
26         if j in num_li:
27
28             if j == i:
29                 bull_cow[0] += 1
30
Enter 4 digit number only. Try again.
Enter your guess: 2345
1 bulls, 1 cows
Enter your guess: 334455
Number should not have repeated digits. Try again.
Enter your guess: 3451
1 bulls, 1 cows
Enter your guess: 3457
0 bulls, 1 cows
Enter your guess: 12345
Enter 4 digit number only. Try again.
Enter your guess: 1234
1 bulls, 2 cows
Enter your guess: 3421
1 bulls, 2 cows
Enter your guess: 2134
2 bulls, 1 cows
Enter your guess: 2143
1 bulls, 2 cows
Enter your guess: 3421
1 bulls, 2 cows
You ran out of tries. Number was 2031
PS F:\c++_course> 2147
2147
PS F:\c++_course> python -u "f:\c++_course\python\project.py"
Enter number of tries: 10
Enter your guess: 2147
0 bulls, 2 cows
Enter your guess: 2134
0 bulls, 1 cows
Enter your guess: 1234
0 bulls, 1 cows
Enter your guess: 4321
1 bulls, 0 cows
Enter your guess: 4235
0 bulls, 1 cows
Enter your guess: 4
```





```
File Edit Selection View Go Run ... c++_course
python_project.ipynb project.py x sliding_window.cpp class.ipynb numpy.ipynb
phyton > project.py > numOfBullsCows
19 def numOfBullsCows(num, guess):
34     return bull_cow
35
36
37
38 num = generateNum()
39 tries = int(input('Enter number of tries: '))
40
41
42 while tries > 0:
43     guess = int(input("Enter your guess: "))
44
45     if not noDuplicates(guess):
46         print("Number should not have repeated digits. Try again.")
47         continue
48     if guess < 1000 or guess > 9999:
49         print("Enter 4 digit number only. Try again.")
50         continue
51
52     bull_cow = numOfBullsCows(num, guess)
53     print(f"{bull_cow[0]} bulls, {bull_cow[1]} cows")
54     tries -= 1
55
56     if bull_cow[0] == 4:
57         print("You guessed right!")
58         break
59 else:
60     print(f"You ran out of tries. Number was {num}")
61
```

```
Enter 4 digit number only. Try again.
Enter your guess: 2345
1 bulls, 1 cows
Enter your guess: 334455
Number should not have repeated digits. Try again.
Enter your guess: 3451
1 bulls, 1 cows
Enter your guess: 3457
0 bulls, 1 cows
Enter your guess: 12345
Enter 4 digit number only. Try again.
Enter your guess: 1234
1 bulls, 2 cows
Enter your guess: 3421
1 bulls, 2 cows
Enter your guess: 2134
2 bulls, 1 cows
Enter your guess: 2143
1 bulls, 2 cows
Enter your guess: 3421
1 bulls, 2 cows
You ran out of tries. Number was 2031
PS F:\c++_course> 2147
2147
PS F:\c++_course> python -u "f:\c++_course\phyton\project.py"
Enter number of tries: 10
Enter your guess: 2147
0 bulls, 2 cows
Enter your guess: 2134
0 bulls, 1 cows
Enter your guess: 1234
0 bulls, 1 cows
Enter your guess: 4321
1 bulls, 0 cows
Enter your guess: 4235
0 bulls, 1 cows
Enter your guess: 4
```

1 4 c++\_course/c++ Debug 0 Share Code Link Explain Code Comment Code Find Bugs Code Chat Blackbox Search Error Port: 3000 Go Live Blackbox Prettier

Links

- The Bulls and Cows game reaches its climax as the player either triumphantly cracks the hidden code within allowed attempts, prompting a congratulatory message with the number of tries taken, or strategically runs out of guesses, leading to the game's reveal of the elusive secret number.

- [GeeksForGeeks](#)
- [Javapoint](#)



# The End