**AMUST Cricket Game - Project Report**

**Description of the Project**

The AMUST Cricket Game is a simple, text-based cricket simulation game developed using the C programming language. The game aims to provide a fun and interactive experience while simulating a short, one-over cricket match. Players can either bat or bowl, depending on the result of a toss, and the game incorporates elements of chance and strategy, making it engaging and unpredictable.

The project demonstrates key programming concepts, including:

* Random number generation for gameplay dynamics.
* User interaction through input/output operations.
* Control structures for decision-making and game flow.

This project is designed for educational purposes and showcases how fundamental programming principles can be applied to create an entertaining game.

**Key Features of the Game**

1. **Toss Mechanism:**
   * The game begins with a toss to decide whether the player or the machine bats first.
   * The machine chooses to bat if it wins the toss, while the player can choose between batting and bowling upon winning the toss.
2. **One-Over Match:**
   * Each inning consists of six balls, offering a concise and fast-paced gaming experience.
3. **Single Wicket Rule:**
   * If a player or the machine is "out," the inning ends immediately.
4. **Machine Batting:**
   * The machine randomly selects a run (0-6) for each ball.
   * The player attempts to guess the machine's run. A correct guess results in the machine being out.
5. **Player Batting:**
   * The player inputs their desired run (0-6) for each ball.
   * The machine attempts to guess the player’s run. A correct guess results in the player being out.
6. **Target Chasing:**
   * The second inning involves chasing a target set by the first inning.
   * The game declares the winner based on the scores at the end of both innings.
7. **Randomization:**
   * The game uses random number generation to simulate the unpredictable nature of cricket, making each playthrough unique.

**Output Details**

**Toss Outcome:**

* Displays whether the player or the machine wins the toss.
* Prompts the player to choose batting or bowling if they win the toss.

**Machine Batting:**

* Displays each ball’s outcome, including the machine's runs and whether the player’s guess was correct.
* Displays the machine's total score after the inning ends.

**Player Batting:**

* Displays each ball’s outcome, including the player's runs and whether the machine’s guess was correct.
* Displays the player’s total score after the inning ends.

**Final Result:**

* Displays the winner (Player or Machine) based on the scores.
* Handles scenarios where the player wins, loses, or ties the game.

**Example Output**

**Toss:**

Machine won the toss and chose to bat first.

**Machine Batting:**

Machine is batting.

Ball 1: Guess the machine's run (0-6): 4

Machine scored 2 runs.

Ball 2: Guess the machine's run (0-6): 3

Out! Player guessed correctly.

Machine's final score: 6

**Player Batting:**

Player is batting to chase a target of 7 runs.

Ball 1: Enter your run (0-6): 3

Player scored 3 runs.

Ball 2: Enter your run (0-6): 2

Player scored 2 runs.

Ball 3: Enter your run (0-6): 1

Out! Machine guessed correctly.

Player loses! Final score: 5

**Final Result:**

Machine wins!

This project is a great demonstration of applying C programming techniques to create a simple yet engaging game. The AMUST Cricket Game combines logical flow, user interaction, and randomness to simulate the excitement of cricket in a compact format.