

KHELA RA SIKA-A Multi Game Program

PRESENTED BY:

ADARSHA BHATTARAI

THA081BCT01

NIKESH BARAL

THA081BCT018

SIRJAN GHIMIRE

THA081BCT043

YUGAL KISHOR ADHIKARI

THA081BCT047

Department of Electronics and Computer

Thapathali Campus

21st March 2025

Introduction

- KHELA RA SIKA is a C-based program that includes multiple games.
- Games included:
 - KBC (Kaun Banega Crorepati)**
 - Hangman**
 - Number Generator.**
- Uses file handling and user interaction to enhance gameplay.
- Designed to provide entertainment and knowledge.

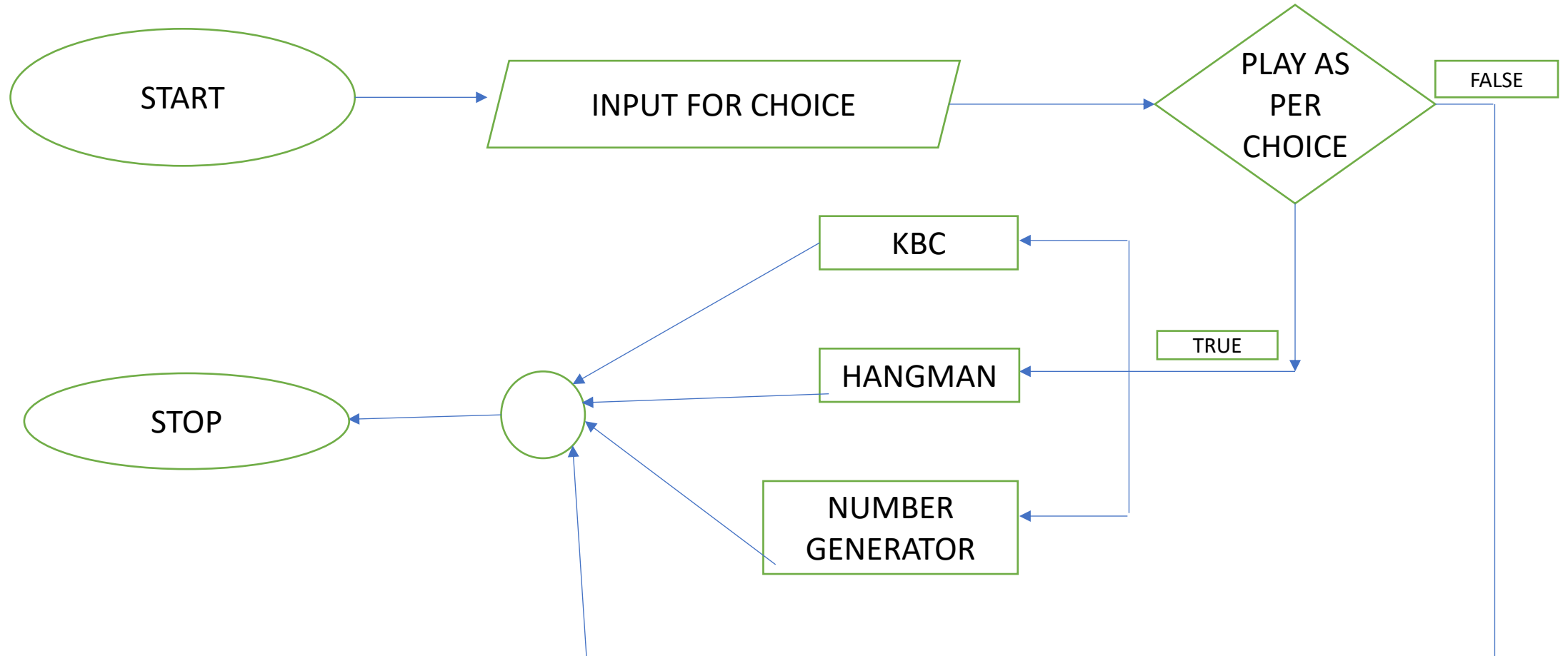
Key Features

- **KBC:**
 - 14 physics questions with lifelines (50/50, Swap).
 - Progressive prize money system.
- **Hangman:**
 - Random word selection from a predefined list.
 - Limited attempts and live feedback.
- **Number Generator:**
 - Random arithmetic problems.
 - Score tracking for correct answers.

Process:

- Main Menu: Users select a game.
- KBC: Quiz with lifelines.
- Hangman: Word guessing game.
- Number Generator: Arithmetic-based game.
- Game loops until win, lose, or quit.

PROGRAM STRUCTURE



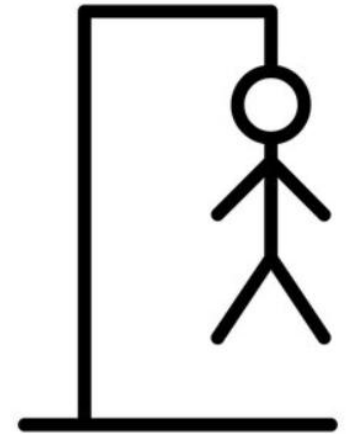
TECHNICAL DETAILS

Code Structure:

- ▶ The program consists of three main functions:
kbcGame(), hangmanGame(), numberGame()
- ▶ Each function operates independently

1 KBC (Kaun Banega Crorepati)

- Arrays: Store questions, options (2D array), and answers (1D array).
- Lifelines:
 - 50/50: Hide two incorrect options via index manipulation.
 - Swap: Replace the current question with a new one using `rand()`.
- Score Tracking: Predefined prize array (e.g., [1000, 2000, ..., 10^6]) mapped to question levels.
- Randomization: `rand()` selects questions from a shuffled pool.



2. Hangman

- Word Selection: Predefined word list; `rand()` picks a word.
- Attempt Limit: Counter decremented on wrong guesses (e.g., 6 attempts).
- Feedback: guessed words and remaining attempts.

3. Number Generator (Math Quiz)

- Problem Generation:
 - `rand()` generates operands (e.g., `num1 = rand() % 100`, `num2 = rand() % 100`).
 - Operator (+, -, *) selected via `rand() % 3`.
- Validation: Compare user input with precomputed results.
- Scoring: Dynamic score update (e.g., +10 per correct answer).

C-COMPONENT IMPLEMENTED

- Function
- String handling
- Pointer
- Array
- Control structure

DIFFICULTY FACED

- CREATING IDEA
- TIME MANAGEMENT
- USING DIFFERENT FUNCTIONS
- UNDERSTANDING FLOW OF PROGRAMS
- ENSURING ACCURACY AND EFFICIENCY



Conclusion & Future Scope

- Successfully created a functional game collection
- Can be used for learning and entertaining purpose
- **FUTURE:**
- Add GUI, include more games, multiplayer support.

THANK YOU!