MINI QUIZ GAME

ప్రగతి

విద్యా పరా దేవతా

Learning is Supreme Deity

TEAM:

25A31A4334	:	SHAIK TAZMEEN RIDA
25A31A4318	:	MOHAMMAD DHANIYA FATHIMA
25A31A4332	:	SABBATHI PAVANI
25A31A4364	:	VANASI GOWTHAM
25A31A4342	:	JANAPAREDDY JASWANTH

Features Required:

1. Display general knowledge questions
2. Provide four multiple-choice options for each question
3. Select questions randomly without repetition
4. Accept user answers and validate them
5. Calculate and display final score

Concepts to be Used: Arrays, Loops, Conditional Statements, Random Number Generation, Standard Input / Output Functions

Technology Used: C Programming Language

Output:

Console-based General Knowledge Quiz

Final score displayed after quiz completion

MINI QUIZ GAME

A C Programming Mini Project

1. Introduction

The General Knowledge Quiz Game is a console-based mini project developed using the C programming language.
The main purpose of this project is to test the general knowledge of users through multiple-choice questions.

The system displays questions one by one, provides four options for each question, and evaluates the user's response.
Questions are selected randomly to ensure that the quiz experience is different every time the program runs.

This project demonstrates the practical application of arrays, loops, and random number generation in C programming.

2. Objectives

- To develop a quiz application using C programming
- To store questions, options, and answers using arrays
- To implement random question selection
- To avoid repetition of questions
- To calculate and display the final score automatically
- To improve logical thinking and programming skills

3. System Features

3.1 Question Display

The system displays one general knowledge question at a time along with four multiple-choice options.
The user must select one option from A, B, C, or D.

3.2 Random Question Selection

Questions are selected randomly using the rand() function.
An additional array is used to track already asked questions, ensuring that no question is repeated during the quiz.

3.3 Answer Validation

The user’s input is compared with the correct answer stored in the array.
If the answer is correct, the score is incremented automatically.

3.4 Score Calculation

After all questions are answered, the system calculates the total score and displays it on the console.

4. Technology & Concepts Used

4.1 C Programming Concepts

- **Arrays** – to store questions, options, and correct answers
- **Loops** – to display multiple questions
- **Conditional Statements** – to check correct answers
- **Random Number Generation** – to select questions randomly

4.2 Randomization Technique

The rand() and srand(time(0)) functions are used to generate random question indices.
An integer array is used to mark already displayed questions.

5. System Design

5.1 Data Structure Used

```
char questions[10][200];
char options[10][4][50];
char correctAnswers[10];
int asked[10];
```

These arrays help in storing quiz data and managing random selection efficiently.

6. Workflow

Main Program Flow:

1. Initialize questions, options, and correct answers
2. Generate a random question index
3. Check if the question is already asked
4. Display the question and options
5. Accept user input
6. Compare input with correct answer
7. Update score
8. Repeat until all questions are completed
9. Display final score

7. Sample Output (Console)

```
GENERAL KNOWLEDGE QUIZ GAME

1. What is the capital of India?
A. Mumbai
B. New Delhi
C. Kolkata
D. Chennai

Your answer: B

QUIZ COMPLETED
Your Score: 8 / 10
```

That number (8) in the *Sample Output* is just an example, not a calculated value from a specific run.

In project reports:

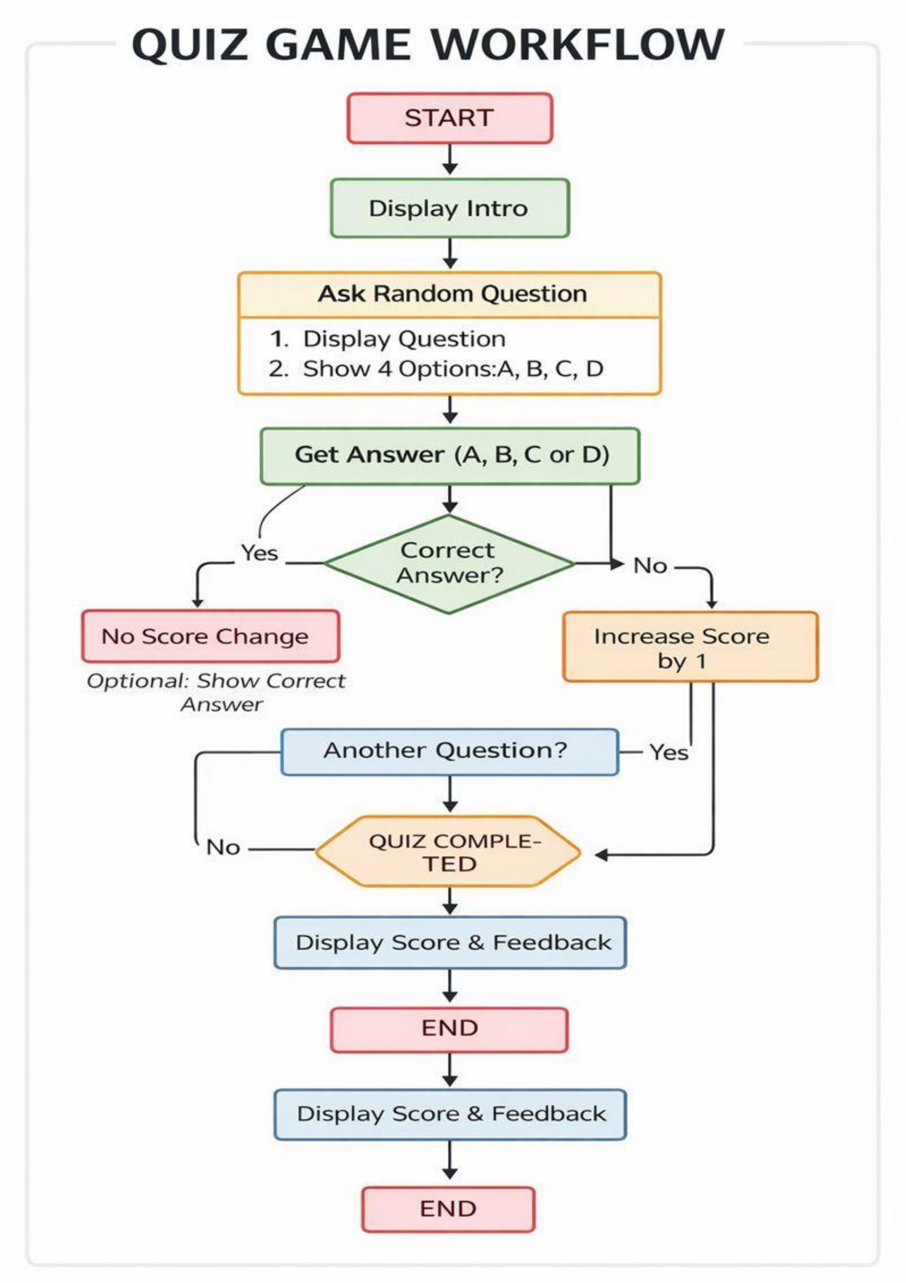
- *Sample Output* shows one possible result
- The score depends on how many questions the user answers correctly
- If the user answers 8 questions correctly → score = 8/10
- If they answer 5 correctly → score = 5/10, and so on

8. Conclusion

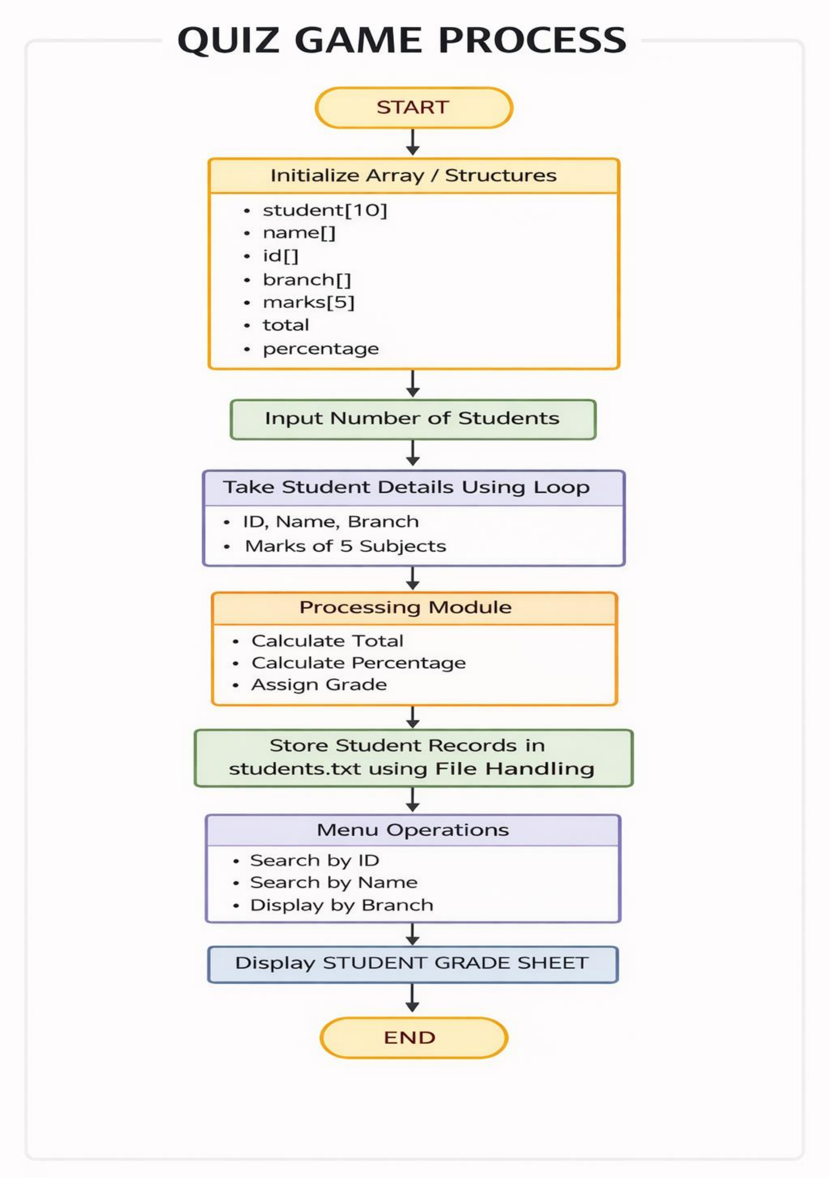
The General Knowledge Quiz Game is a simple yet effective C-based mini project.
It demonstrates the use of arrays, loops, conditional statements, and random number generation.

The project improves logical thinking and provides hands-on experience with structured programming concepts.
It is suitable for beginners and can be extended further by adding difficulty levels, timers, or file handling.

WORK FLOW


QUIZ GAME WORKFLOW
START
Display Intro
Ask Random Question
1. Display Question
2. Show 4 Options:A, B, C, D
Get Answer (A, B, C or D)
Yes
Correct Answer?
No
No Score Change
Optional: Show Correct Answer
Increase Score by 1
Another Question?
Yes
No
QUIZ COMPLE-TED
Display Score & Feedback
END
Display Score & Feedback
END

PROCESS



QUIZ GAME PROCESS

START

Initialize Array / Structures

- student[10]
- name[]
- id[]
- branch[]
- marks[5]
- total
- percentage

Input Number of Students

Take Student Details Using Loop

- ID, Name, Branch
- Marks of 5 Subjects

Processing Module

- Calculate Total
- Calculate Percentage
- Assign Grade

Store Student Records in students.txt using File Handling

Menu Operations

- Search by ID
- Search by Name
- Display by Branch

Display STUDENT GRADE SHEET

END

SAMPLE CODE

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>



int main() {

    int i, score = 0;

    char answer;

    int asked[10] = {0};
// to avoid repeating
questions

    int q;



    // Questions

    char
questions[10][200] =
{

        "What is the
capital of India?",

        "Which planet is
known as the Red
Planet?",
        "Who is known
as the Father of the
Indian
Constitution?",

        "Which is the
largest ocean in the
world?",

        "What is the
national animal of
India?",

        "Which gas is
most abundant in
Earth's
atmosphere?",

        "Who invented
the telephone?",

        "Which is the
longest river in the
world?",

        "Which country
is known as the Land
of the Rising Sun?",

        "What is the SI
unit of force?"

    };
    // Options

    char
options[10][4][50] = {

        {"Mumbai",
"New Delhi",
"Kolkata",
"Chennai"},

        {"Earth",
"Venus", "Mars",
"Jupiter"},

        {"Mahatma
Gandhi", "Jawaharlal
Nehru", "B. R.
Ambedkar",
"Rajendra Prasad"},

        {"Indian Ocean",
"Atlantic Ocean",
"Arctic Ocean",
"Pacific Ocean"},

        {"Lion",
"Elephant", "Tiger",
"Leopard"},

        {"Oxygen",
"Nitrogen", "Carbon
dioxide",
"Hydrogen"},
        {"Thomas
Edison", "Alexander
Graham Bell", "Nikola
Tesla", "James
Watt"},

        {"Amazon",
"Yangtze",
"Mississippi", "Nile"},

        {"China",
"Thailand", "Japan",
"South Korea"},

        {"Joule",
"Pascal", "Newton",
"Watt"}

    };



    // Correct answers

    char
correctAnswers[10] =
{'B','C','C','D','C','B','B'
,'D','C','C'};



    srand(time(0));   //
seed for randomness
```

```
    printf("GENERAL
KNOWLEDGE QUIZ
GAME\n");
    printf("Questions
will appear in
random order\n");
    printf("Choose A,
B, C or D\n\n");


    for(i = 0; i < 10; i++)
{


        // generate a
random unused
question index
        do {
            q = rand() %
10;
        } while(asked[q]
== 1);


        asked[q] = 1;
        printf("%d.
%s\n", i + 1,
questions[q]);

        printf("A. %s\n",
options[q][0]);

        printf("B. %s\n",
options[q][1]);

        printf("C. %s\n",
options[q][2]);

        printf("D. %s\n",
options[q][3]);



        printf("Your
answer: ");

        scanf(" %c",
&answer);



        if(answer ==
correctAnswers[q] ||
answer ==
correctAnswers[q] +
32)

            score++;
        printf("\n");
    }

    printf("QUIZ COMPLETED\n");
    printf("Your Score: %d / 10\n", score);

    if(score >= 8)
        printf("Excellent performance\n");
    else if(score >= 5)
        printf("Good performance\n");
    else
        printf("Needs improvement\n");

    return 0;
}
```