CMR UNIVERSITY
Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

A Project Report
on

“ QUIZ GAME ”

Subject: Problem Solving through Programming in C
Subject code: 4BCS105/205
Semester: Ist Semester

Submitted in partial fulfillment of the requirements for the award of degree in

Bachelor of Technology

in

Computer Science and Engineering

Submitted by:

KRITIK AGARWAL (19BBTCS067)

Under the Guidance of:
Mrs. S. Vanitha
Assistant Professor
Dept. of CSE, SOET

Department of Computer Science and Engineering

Off Hennur - Bagalur Main Road,

Near Kempegowda International Airport, Chagalahatti,

Bangalore, Karnataka-562149

2019-2020

CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work, entitled " QUIZ GAME ", submitted to the CMR University, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a record of work done by Mr. Kritik Agarwal bearing university register number 19BBTCS067 during the academic year 2019-20 at School of Engineering and Technology, CMR University, Bangalore under my supervision and guidance. The project report has been approved as it satisfies the academic requirement in respect of project course prescribed in the 1st semester, in the said degree.

Signature of the Guide	Signature of the HOD	Signature of the Dean
Mrs. S. Vanitha	Dr. Arun Biradar	Dr. Jayaprasad M
Assistant Professor	Professor and Head	Professor and Dean
Dept. of CSE, SOET, CMRU	Dept. of CSE, SOET, CMRU	SOET, CMRU

Examiner Signature with date

CMR UNIVERSITY

Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

DECLARATION

I, KRITIK AGARWAL (Reg. No. 19BBTCS067) student of 1st semester B.Tech. Computer Science and Engineering, School of Engineering and Technology, Bangalore, hereby declare that the project work entitled " QUIZ GAME " has been carried out by me under the guidance of Mrs. S VANITHA , Assistant Professor, Department of Computer Science and Engineering, School of Engineering and Technology. This report is submitted in partial fulfillment of the requirement for award of Bachelor of Technology in Computer Science and Engineering by CMR University, Bangalore, during the academic year 2019-2020. The project report has been submitted as it satisfies the academic requirement in respect of project course prescribed in the 1st semester, in the said degree.

Place: Bangalore

Date:

KRITIK AGARWAL (19BBTCS067)

CMR UNIVERSITY
Private University Established in Karnataka State by Act No. 45 of 2013

SCHOOL OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the friends and guide, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of the project. We express our heartfelt sincere gratitude to Dr. Jayaprasad M, Dean, School of Engineering and Technology, CMR University for his support. We would like to express our thanks to Dr. Arun Biradar, Professor and Head, Department of Computer Science and Engineering, School of Engineering and Technology, CMR University, Bangalore, for his encouragement that motivated us for the successful completion of project work. We express our thanks to our Project Guide Mrs. S. Vanitha, Assistant Professor, Department of Computer Science and Engineering, School of Engineering and Technology, CMR University for her constant support.

KRITIK AGARWAL (19BBTCS067)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Quiz Game Project is a simple console application designed to demonstrate the practical use of C programming language and its features.

Quiz is basically a game where a question is asked and 4 options are provided to the player. Any one of the 4 option is the correct answer. The player as to guess the correct one.

In this system, first the menu option is displayed which says “Play Game or Quit”. Upon selecting “Play Game” the user has to enter their name and then the instructions are displayed. Then the user is transferred to Selection Round where they are asked three questions minimum two correct answers are necessary for passing. If passed, then only user can proceed for the quiz questions. In Quiz Questions, if the user answers correctly, 1 point is awarded. And after the completion of the game, it displays “Game Over” along with the user’s name and score.

This system can be used in Schools/Colleges or vice versa for conducting Quiz contest.

CHAPTER 1

INTRODUCTION TO PROBLEM SOLVING THROUGH PROGRAMMING

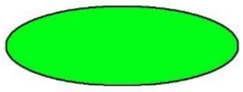
1.1 INTRODUCTION

FLOWCHART

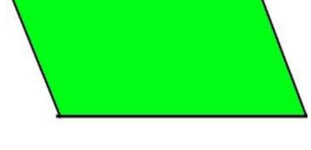
Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting".

Basic Symbols used in Flowchart Designs

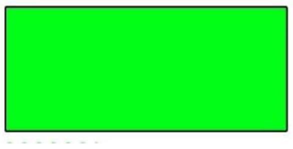
1. **Terminal:** The oval symbol indicates Start, Stop and Halt in a program's logic flow. A pause/halt is generally used in a program logic under some error conditions. Terminal is the first and last symbols in the flowchart.



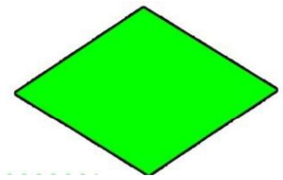
2. **Input/Output:** A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.



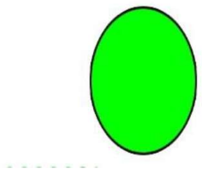
3. **Processing:** A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.



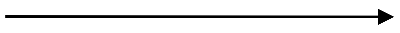
4. **Decision:** Diamond symbol represents a decision point. Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.



5. **Connectors:** Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



6. **Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.



PSEUDO CODE

Pseudo code: It’s simply an implementation of an algorithm in the form of annotations and informative text written in plain English. It has no syntax like any of the programming language and thus can’t be compiled or interpreted by the computer.

Advantages of Pseudocode

- Improves the readability of any approach. It’s one of the best approaches to start implementation of an algorithm.

- Acts as a bridge between the program and the algorithm or flowchart. Also works as a rough documentation, so the program of one developer can be understood easily when a pseudo code is written out. In industries, the approach of documentation is essential. And that’s where a pseudo-code proves vital.

- The main goal of a pseudo code is to explain what exactly each line of a program should do, hence making the code construction phase easier for the programmer.

EXAMPLE-

```
BEGIN
FUNCTION Welcome( )
FUNCTION menu( )
END
```

ALGORITHM

An algorithm in general is a sequence of steps to solve a particular problem. Algorithms are universal. The algorithm you use in C programming language is also the same algorithm you use in every other language. The only difference is the syntax. Each language has its own syntax. Let's say we need an algorithm to go to class when we wake up. We can write it as thus:

STEP 1-Wake up.

STEP 2-Brush teeth and bath.

STEP 3-Wear uniform.

STEP 4-Eat breakfast.

STEP 5-Take transport to school.

STEP 6-Get to class.

Notice how some steps can be omitted (step 4) or some other steps can be added. The point is this is a basic structure of how to get to school i.e sequence of steps to solve the problem; how to get to class when I wake up.

In step 5, some people make take the school bus as transportation others may walk. That is where the syntax of a programming language comes into play. Though we get step 5 done.

1.2 CONDITIONAL BRANCHING AND ITERATIVE LOOPS

Conditional statements help you to make a decision based on certain conditions. These conditions are specified by a set of conditional statements having boolean expressions which are evaluated to a boolean value true or false. There are following types of conditional statements in C.

1. If statement
2. If-Else statement
3. Nested If-else statement
4. If-Else If ladder
5. Switch statement
6. Goto statement

1.2.1 IF STATEMENT

The single if statement in C language is used to execute the code if a condition is true. It is also called one-way selection statement.

Syntax

```
if(expression)
{
 //code to be executed
}
```

1.2.2 IF-ELSE STATEMENT

The if-else statement in C language is used to execute the code if condition is true or false. It is also called two-way selection statement.

Syntax

```
if(expression)
{
 //Statements
}
else
{
 //Statements
}
```

1.2.3 NESTED IF-ELSE STATEMENT

The nested if...else statement is used when a program requires more than one test expression. It is also called a multi-way selection statement. When a series of the decision are involved in a statement, we use if else statement in nested form.

Syntax:

```
if( expression )
{
 if( expression1 )
 {
 statement-block1;
 }
 else
 {
 statement-block 2;
 }
}
else
{
 statement-block 3;
}
```

1.2.4 IF-ELSE IF LADDER

The if-else-if statement is used to execute one code from multiple conditions. It is also called multipath decision statement. It is a chain of if..else statements in which each if statement is associated with else if statement and last would be an else statement.

Syntax

```
if(condition1)
{
 //statements
}
else if(condition2)
{
 //statements
}
else if(condition3)
{
 //statements
}
else
{
 //statements
}
```

1.2.5 SWITCH STATEMENT

switch statement acts as a substitute for a long if-else-if ladder that is used to test a list of cases. A switch statement contains one or more case labels which are tested against the switch expression. When the expression match to a case then the associated statements with that case would be executed.

Syntax

```
Switch (expression)
{
 case value1:
 //Statements
 break;
 case value 2:
 //Statements
 break;
 case value 3:
 //Statements
 case value n:
 //Statements
 break;
 Default:
 //Statements
}
```

Note:

- The switch statement must be an integral type.
- Case labels must be constants.
- Case labels must be unique.
- Case labels must end with a colon.
- The break statement transfers the control out of switch statement.
- The break statement is optional.

1.2.6 GOTO STATEMENT

A **goto** statement in C programming provides an unconditional jump from the 'goto' to a labeled statement in the same function.

Syntax:

```
goto label;
..
.
label: statement;
```

Here **label** can be any plain text except C keyword and it can be set anywhere in the C program above or below to **goto** statement.

1.2.7 LOOPS IN C

Types of Loop

There are 3 types of Loop in C language, namely:

1. while loop
2. for loop
3. do while loop

1.2.7.1 WHILE LOOP

while loop can be addressed as an **entry control** loop. It is completed in 3 steps.

- Variable initialization. (e.g int x = 0;)
- Condition (e.g while(x <= 10))
- Variable increment or decrement (x++ or x-- or x = x + 2)

Syntax:

```
variable initialization;
while(condition)
{
   statements;
   variable increment or decrement;
}
```

1.2.7.2 FOR LOOP

for loop is used to execute a set of statements repeatedly until a particular condition is satisfied. We can say it is an **open ended loop.**. General format is,

```
for(initialization; condition; increment/decrement)
{
   statement-block;
}
```

In for loop we have exactly two semicolons, one after initialization and second after the condition. In this loop we can have more than one initialization or increment/decrement, separated using comma operator. But it can have only one **condition**.

The for loop is executed as follows:

1. It first evaluates the initialization code.
2. Then it checks the condition expression.
3. If it is **true**, it executes the for-loop body.
4. Then it evaluate the increment/decrement condition and again follows from step 2.
5. When the condition expression becomes **false**, it exits the loop.

1.2.7.3 NESTED FOR LOOP

We can also have nested for loops, i.e one for loop inside another for loop. Basic syntax is,

```
for(initialization; condition; increment/decrement)
{
   for(initialization; condition; increment/decrement)
   {
      statement ;
   }
}
```

1.2.7.4 DO WHILE LOOP

In some situations it is necessary to execute body of the loop before testing the condition. Such situations can be handled with the help of do-while loop. do statement evaluates the body of the loop first and at the end, the condition is checked using while statement. It means that the body of the loop will be executed at least once, even though the starting condition inside while is initialized to be **false**. General syntax is,

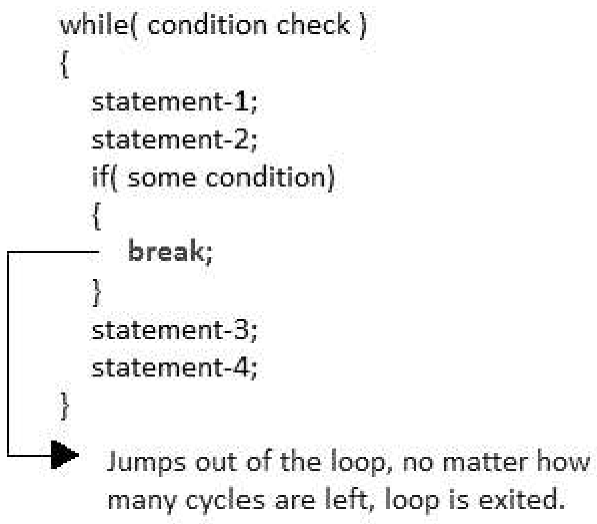
```
do
{
   .....
   .....
}
while(condition)
```

1.2.8 JUMPING OUT OF LOOPS

Sometimes, while executing a loop, it becomes necessary to skip a part of the loop or to leave the loop as soon as certain condition becomes **true**. This is known as jumping out of loop.

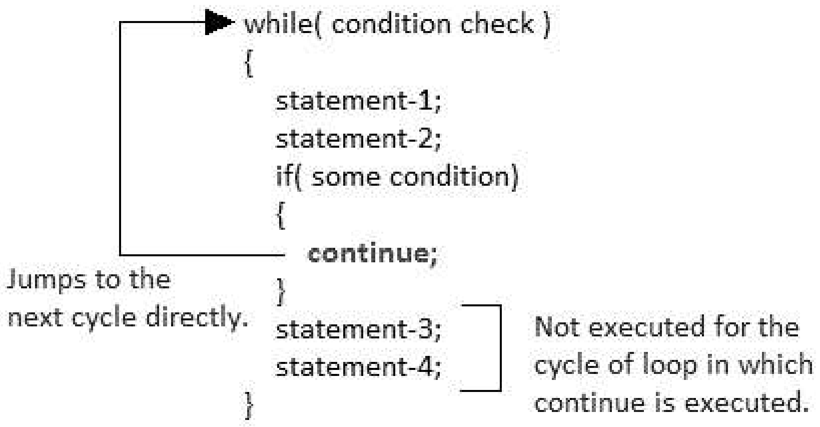
1.2.8.1 break statement

When break statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.


```
while( condition check )
{
   statement-1;
   statement-2;
   if( some condition)
   {
      break;
   }
   statement-3;
   statement-4;
}
```
Jumps out of the loop, no matter how many cycles are left, loop is exited.

1.2.8.2 continue statement

It causes the control to go directly to the test-condition and then continue the loop process. On encountering continue, cursor leave the current cycle of loop, and starts with the next cycle.


```
while( condition check )
{
   statement-1;
   statement-2;
   if( some condition)
   {
      continue;
   }
   statement-3;
   statement-4;
}
```
Jumps to the next cycle directly.
Not executed for the cycle of loop in which continue is executed.

1.3 ARRAYS & STRINGS

1.3.1 Array

An array is a variable that can store multiple values. For example, if we want to store 100 integers, you can create an array for it.

```
int data[100];
```

ARRAY DECLARATION:

```
dataType arrayName[arraySize];
```

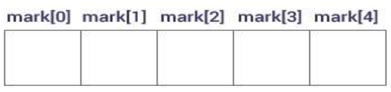
For example,

```
float mark[5];
```

Here, we declared an array, mark, of floating-point type. And its size is 5. Meaning, it can hold 5 floating-point values.
It's important to note that the size and type of an array cannot be changed once it is declared.

Accessing Array Elements

We can access elements of an array by indices.
Suppose we declared an array mark as above. The first element is mark[0], the second element is mark[1] and so on.



INITIALISING AN ARRAY

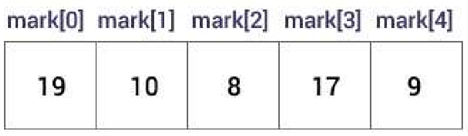
It is possible to initialize an array during declaration. For example,

```
int mark[5] = {19, 10, 8, 17, 9};
```

We can also initialize an array like this.

```
int mark[] = {19, 10, 8, 17, 9};
```

Here, we haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.



Here,

```
mark[0] is equal to 19
mark[1] is equal to 10
mark[2] is equal to 8
mark[3] is equal to 17
mark[4] is equal to 9
```

Changing Value of Array elements

```
int mark[5] = {19, 10, 8, 17, 9}
// make the value of the third element to -1
mark[2] = -1;
// make the value of the fifth element to 0
mark[4] = 0;
```

Input and Output Array Elements

Here's how we can take input from the user and store it in an array element.

```
// take input and store it in the 3rd element
scanf("%d", &mark[2]);
// take input and store it in the ith element
scanf("%d", &mark[i-1]);
```

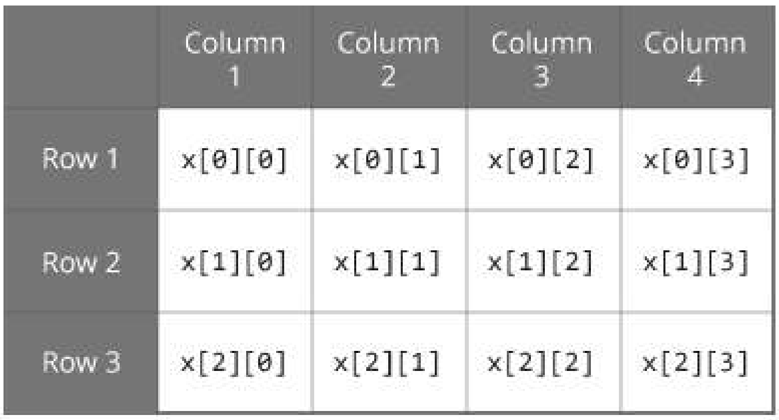
Here's how you can print an individual element of an array.

```
// print the first element of the array
printf("%d", mark[0]);
// print the third element of the array
printf("%d", mark[2]);
// print ith element of the array
printf("%d", mark[i-1]);
```

1.3.2 2D Array

In C programming, we can create an array of arrays. These arrays are known as multidimensional arrays. For example, float x[3][4];

Here, x is a two-dimensional (2d) array. The array can hold 12 elements. We can think the array as a table with 3 rows and each row has 4 columns.



	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Initialization of a 2d array

Different ways to initialize two-dimensional array:

1. int c[2][3] = {{1, 3, 0}, {-1, 5, 9}};
2. int c[][3] = {{1, 3, 0}, {-1, 5, 9}};
3. int c[2][3] = {1, 3, 0, -1, 5, 9};

1.3.3 String

String is an array of characters.

String Declaration

Method 1:

char address[]={'T', 'E', 'X', 'A', 'S', '\0'};

Method 2: The above string can also be defined as –

char address[]="TEXAS";

In the above declaration NULL character (\0) will automatically be inserted at the end of the string.

NULL Char “\0”

'\0' represents the end of the string. It is also referred as String terminator & Null Character.

String I/O in C programming

Reading & writing Strings in C using printf() and scanf() functions

```
#include <stdio.h>

#include <string.h>

int main()

{

   /* String Declaration*/

   char nickname[20];

   printf("Enter your Nick name:");

   /* I am reading the input string and storing it in nickname. Array name alone works as a
base address of array so, we can use “nickname” instead of “&nickname” here */

   scanf("%s", nickname);

   /*Displaying String*/

   printf("%s",nickname);

   return 0;

}
```

Note: “%s” format specifier is used for strings input/output.

Reading & Writing Strings in C using gets() and puts() functions

```
#include <stdio.h>

#include <string.h>

int main()

{

   /* String Declaration*/

   char nickname[20];

   /* Console display using puts */

   puts("Enter your Nick name:");

   /*Input using gets*/

   gets(nickname);

   puts(nickname);

   return 0;

}
```

Also, gets() is an old function.

The new alternative for it is fgets(variable_name, sizeof(variable_name),stdin);

1.3.3 String functions

We can perform functions to compare two strings, concatenate strings, copy one string to another & perform various string manipulation operations using the pre-defined functions of "string.h" header file. In order to use these string functions we must include string.h file in your C program.

String function – strlen

Syntax:

```
strlen(str);
```

It returns the length of the string without including end character (terminating char '\0')..

String function – strcmp

Syntax:

```
strcmp(str1, str2);
```

It compares the two strings and returns an integer value. If both the strings are same (equal) then this function would return 0 otherwise it may return a negative or positive value based on the comparison.

If string1 < string2 OR string1 is a substring of string2 then it would result in a negative value. If string1 > string2 then it would return positive value.

If string1 == string2 then you would get 0(zero) when you use this function for compare strings.

String function – strcat

Syntax:

```
strcat(str1, str2);
```

It concatenates two strings and returns the concatenated string.

String function – strcpy

Syntax:

```
strcpy(str1, str2);
```

It copies the string str2 into string str1, including the end character (terminator char '\0').

String function – strrev

Syntax:

```
strrev(string);
```

It reverses the string.

1.4 FUNCTIONS

A function is a set of statements that may take inputs, do some specific computation and produces output.

The idea is to put some commonly or repeatedly done task together and make a function so that instead of writing the same code again and again for different inputs, we can call the function.

Functions provide abstraction. For example, we can use library functions without worrying about their internal working

Types of function

There are two types of function in C programming:

1. Standard library functions
2. User-defined function

Standard library functions

The standard library functions are built-in functions in C programming. These functions are defined in header files.

For example, the printf() is a standard library function to send formatted output to the screen (display output on the screen). This function is defined in the “stdio.h” header file.

Hence, to use the printf() function, we need to include the stdio.h header file using

#include <stdio.h>

User-defined function

We can also create functions as per our need. Such functions created by the user are known as user-defined functions.

How user-defined function works…

```
#include <stdio.h>

void functionName()

{

    ... .. ...

    ... .. ...

}

int main()

{

    ... .. ...

    ... .. ...

    functionName();

    ... .. ...

    ... .. ...

}
```

The execution of a C program begins from the main() function.

When the compiler encounters functionName(); , control of the program jumps to

void functionName()

And, the compiler starts executing the codes inside functionName().

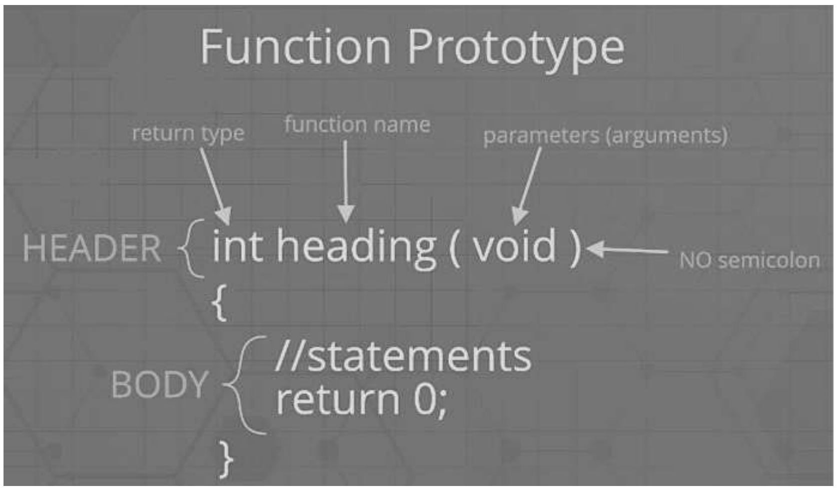
The control of the program jumps back to the main() function once code inside the function definition is executed.

Advantages of user-defined function

- Functions help us in reducing code redundancy. If functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This also helps in maintenance as we have to change at one place if we make future changes to the functionality.
- Functions make code modular. Consider a big file having many lines of codes. It becomes really simple to read and use the code if the code is divided into functions.
- The program will be easier to understand, maintain and debug.
- Reusable codes that can be used in other programs
- A large program can be divided into smaller modules. Hence, a large project can be divided among many programmers

Function Declaration

A function declaration tells the compiler about the number of parameters function takes, data-types of parameters and return type of function. Putting parameter names in function declaration is optional in the function declaration, but it is necessary to put them in the definition.



It is always recommended to declare a function before it is used.

In C, we can do both declaration and definition at the same place.

C also allows to declare and define functions separately, this is especially needed in case of library functions. The library functions are declared in header files and defined in library files.

Parameter Passing to functions

The parameters passed to function are called actual parameters. For example, in the above program 10 and 20 are actual parameters.

The parameters received by function are called formal parameters. For example, in the above program x and y are formal parameters.

There are two most popular ways to pass parameters.

1. Call by Value
2. Call by Reference

Call By Value

In this parameter passing method, values of actual parameters are copied to function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of caller.

Call By Reference

Both actual and formal parameters refer to same locations, so any changes made inside the function are actually reflected in actual parameters of caller.

In C, parameters are always passed by value. Parameters are always passed by value in C.

1.5 STRUCTURES, UNION, FILE HANDLING and ENUMERATION.

1.5.1 C Structures

Structures are collection of dissimilar datatypes.

Defining a structure

“struct” keyword is used to define a structure.

struct defines a new data type which is a collection of primary and derived datatypes.

Syntax:

```
struct [structure_tag]

{

   //member variable 1

   //member variable 2

   //member variable 3

   ...

}[structure_variables];
```

We start with the struct keyword, then it's optional to provide our structure a name, we suggest to give it a name, then inside the curly braces, we have to mention all the member variables, which are nothing but normal C language variables of different types like int, float, array etc.

After the closing curly brace, we can specify one or more structure variables, again this is optional.

Note: The closing curly brace in the structure type declaration must be followed by a semicolon(;).

1) **<u>Declaring Structure variables separately</u>**

```
struct Student

{

   char name[25];

   int age;

   char branch[10];

   //F for female and M for male

   char gender;

};

struct Student S1, S2;      //declaring variables of struct Student
```

2) **<u>Declaring Structure variables with structure definition</u>**

```
struct Student

{

   char name[25];

   int age;

   char branch[10];

   //F for female and M for male

    char gender;
}S1, S2;
```

Here S1 and S2 are variables of structure Student.

However, this approach is not much recommended

Accessing Structure Members

Structure members can be accessed and assigned values in a number of ways. Structure members have no meaning individually without the structure. In order to assign a value to any structure member, the member name must be linked with the structure variable using a dot "." operator also called period or member access operator.

For example:

```
#include<stdio.h>

#include<string.h>

struct Student

{

   char name[25];

   int age;

   char branch[10];

   //F for female and M for male

   char gender;

};

int main()

{
   struct Student s1;

   //s1 is a variable of Student type and age is a member of Student

   s1.age = 18;

     //using string function to add name

   strcpy(s1.name, "Viraaj");

      //displaying the stored values

   printf("Name of Student 1: %s\n", s1.name);

   printf("Age of Student 1: %d\n", s1.age);

   return 0;

}
```

We can also use scanf() to give values to structure members through terminal.

```
scanf(" %s ", s1.name);

scanf(" %d ", &s1.age);
```

Structure Initialization

Like a variable of any other datatype, structure variable can also be initialized at compile time.

```
struct Patient

{

    float height;

    int weight;
    int age;

};

struct Patient p1 = { 180.75 , 73, 23 };    //initialization

or,

struct Patient p1;

p1.height = 180.75;     //initialization of each member separately

p1.weight = 73;

p1.age = 23;
```

Array of Structure

We can also declare an array of structure variables. in which each element of the array will represent a structure variable. Example: struct employee emp[5];

The below program defines an array emp of size 5. Each element of the array emp is of type Employee.

```
#include<stdio.h>

struct Employee

{

    char ename[10];

    int sal;

};

struct Employee emp[5];

int i, j;
```

```
void ask()

{

     for(i = 0; i < 3; i++)

    {

        printf("\nEnter %dst Employee record:\n", i+1);

        printf("\nEmployee name:\t");

        scanf("%s", emp[i].ename);

        printf("\nEnter Salary:\t");

        scanf("%d", &emp[i].sal);

    }

    printf("\nDisplaying Employee record:\n");

    for(i = 0; i < 3; i++)

    {

        printf("\nEmployee name is %s", emp[i].ename);

        printf("\nSlary is %d", emp[i].sal);

    }

}

void main()

{

    ask();
}
```

Structure as Function Arguments

We can pass a structure as a function argument just like we pass any other variable or an array as a function argument.

Example:

```
#include<stdio.h>

struct Student

{

    char name[10];

    int roll;

};

void show(struct Student st);

void main()

{

    struct Student std;

    printf("\nEnter Student record:\n");

    printf("\nStudent name:\t");

    scanf("%s", std.name);

    printf("\nEnter Student rollno.:\t");

    scanf("%d", &std.roll);
```

```
   show(std);

}

void show(struct Student st)

{

   printf("\nstudent name is %s", st.name);

   printf("\nroll is %d", st.roll);

}
```

1.5.2 C Unions

A union is a special data type available in C that allows to store different data types in the same memory location.

We can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple-purpose.

Defining a Union

To define a union, we must use the union statement in the same way as we did while defining a structure. The union statement defines a new data type with more than one member for our program.

The format of the union statement is as follows −

```
union [union tag]

{

  member definition;

  member definition;
```

```
   ...

   member definition;

} [one or more union variables];
```

The union tag is optional and each member definition is a normal variable definition, such as int i; or float f; or any other valid variable definition. At the end of the union's definition, before the final semicolon, we can specify one or more union variables but it is optional. Here is the way we would define a union type named Data having three members i, f, and str –

```
union Data

{

   int i;

   float f;

   char str[20];

} data;
```

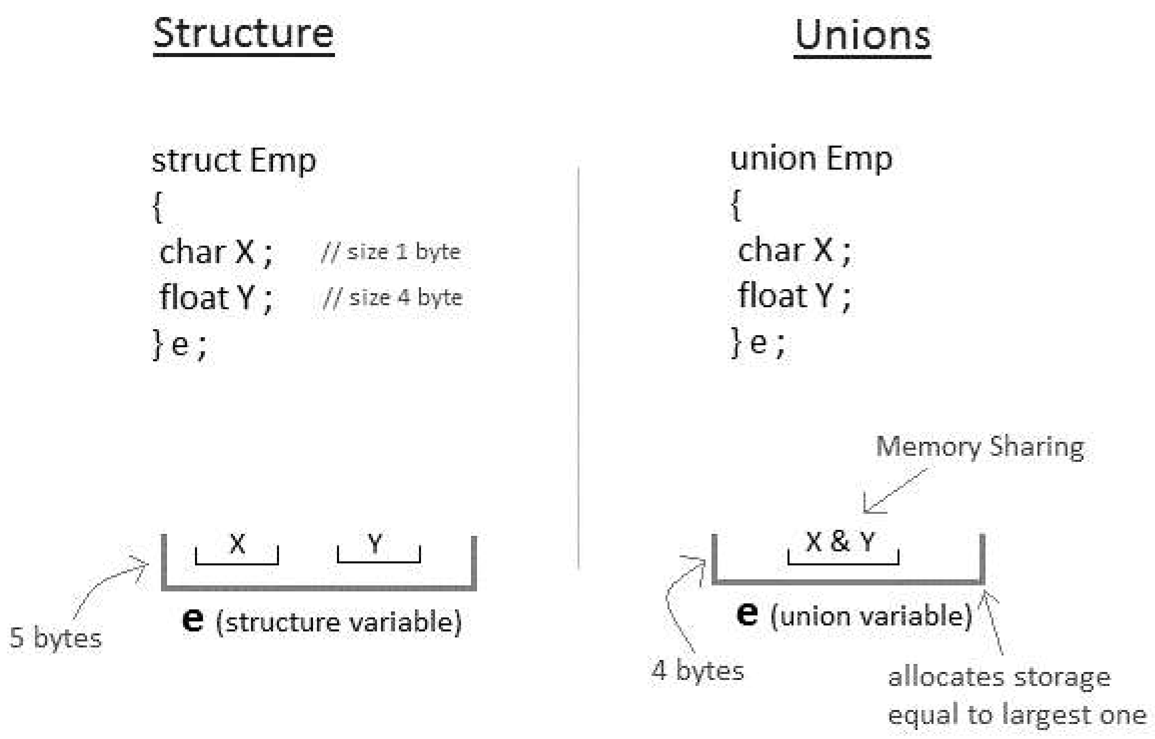
Now, a variable of Data type can store an integer, a floating-point number, or a string of characters. It means a single variable, i.e., same memory location, can be used to store multiple types of data. We can use any built-in or user defined data types inside a union based on our requirement.

The memory occupied by a union will be large enough to hold the largest member of the union. For example, in the above example, Data type will occupy 20 bytes of memory space because this is the maximum space which can be occupied by a character string.

Accessing Union Members

To access any member of a union, we use the member access operator (.). The member access operator is coded as a period between the union variable name and the union member that we wish to access. You would use the keyword union to define variables of union type.

Difference Between Union & Structure



1.5.3 File Handling in C

So far the operations using C program are done on a prompt / terminal which is not stored anywhere. But in the software industry, most of the programs are written to store the information fetched from the program. One such way is to store the fetched information in a file. Different operations that can be performed on a file are:

- Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w++”)
- Opening an existing file (fopen)

- Reading from file (fscanf or fgetc)
- Writing to a file (fprintf or fputs)
- Moving to a specific location in a file (fseek, rewind)
- Closing a file (fclose)

Opening or creating file

For opening a file, fopen function is used with the required access modes. Some of the commonly used file access modes are mentioned below.

File opening modes in C:

- "r" – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the first character in it. If the file cannot be opened fopen( ) returns NULL.

- "w" – Searches file. If the file exists, its contents are overwritten. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.

- "a" – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.

- "r+" – Searches file. If is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the first character in it. Returns NULL, if unable to open the file.

- "w+" – Searches file. If the file exists, its contents are overwritten. If the file doesn't exist a new file is created. Returns NULL, if unable to open file.

- "a+" – Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer which points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open file.

As given above, if we want to perform operations on a binary file, then you have to append ‘b’ at the last. For example, instead of “w”, we have to use “wb”, instead of “a+” we have to use “a+b”. For performing the operations on the file, a special pointer called File pointer is used which is declared as “FILE *filePointer;”

So, the file can be opened as

filePointer = fopen(“fileName.txt”, “w”)

The second parameter can be changed to contain all the attributes listed in the above table.

Reading from a file

The file read operations can be performed using functions fscanf or fgets. Both the functions performed the same operations as that of scanf and gets but with an additional parameter, the file pointer. So, it depends on us if you want to read the file line by line or character by character.

And the code snippet for reading a file is as:

```
FILE * filePointer;

filePointer = fopen(“fileName.txt”, “r”);

fscanf(filePointer, "%s %s %s %d", str1, str2, str3, &year);
```

Writing a file

The file write operations can be perfomed by the functions fprintf and fputs with similarities to read operations. The snippet for writing to a file is as :

```
FILE *filePointer ;

filePointer = fopen(“fileName.txt”, “w”);

fprintf(filePointer, "%s %s %s %d", "We", "are", "in", 2012);
```

Closing a file

After every successful file operations, we must always close a file. For closing a file, we have to use fclose function. The snippet for closing a file is given as :

```
 FILE *filePointer ;

filePointer= fopen("fileName.txt", "w");

---------- Some file Operations -------

fclose(filePointer);
```

1.5.4 Enumeration

Enumeration (or enum) is a user defined data type in C.

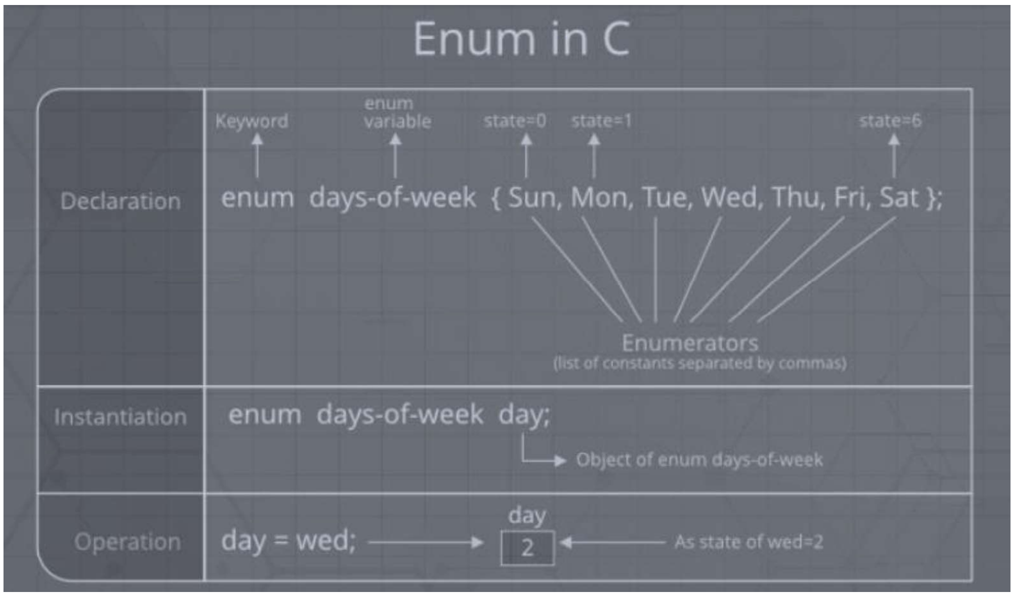
It is mainly used to assign names to integral constants, the names make a program easy to read and maintain.

The keyword 'enum' is used to declare new enumeration types in C.

Variables of type enum can also be defined.

They can be defined in two ways:

```
// In both of the below cases, "day" is defined as the variable of type week.

enum week{Mon, Tue, Wed};

enum week day;

          Or

enum week{Mon, Tue, Wed}day;
```



FACTS about initialization of enum.

1. Two enum names can have same value.
2. If we do not explicitly assign values to enum names, the compiler by default assigns values starting from 0.
3. We can assign values to some name in any order. All unassigned names get value as value of previous name plus one.
4. The value assigned to enum names must be some integeral constant, i.e., the value must be in range from minimum possible integer value to maximum possible integer value.
5. All enum constants must be unique in their scope.

1.6 OVERVIEW

- C is a procedural programming language.
- It was developed in the year 1972 by Dennis Ritchie as a system programming language to write an operating system.
- The main features of C language include low-level access to memory, a simple set of keywords, and clean style, these features make C language suitable for system programmings like an operating system or compiler development.
- Many later languages have borrowed syntax/features directly or indirectly from C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on C language.
- It is a highly portable language which means programs written in 'C' language can run on other machines. This feature is essential if we wish to use or execute the code on another computer
- 'C' is a structured programming language in which program is divided into various modules. Each module can be written separately and together it forms a single 'C' program. This structure makes it easy for testing, maintaining and debugging processes.
- 'C' contains 32 keywords, various data types and a set of powerful built-in functions that make programming very efficient.
- Another feature of 'C' programming is that it can extend itself. A 'C' program contains various functions which are part of a library. We can add our features and functions to the library. We can access and use these functions anytime we want in our program. This feature makes it simple while working with complex programming

1.7 PROBLEM STATEMENT

As schools/colleges or any place where there is a need to conduct quiz, they need to first prepare sets of papers, get it printed according to the number of students/participants and then after conduction ensure smooth manual correction and record to see who scored highest marks.

So, this software "Quiz Game" will reduce the time taken for all this and also the labour cost. And also, the scores are calculated automatically without any error.

1.8 OBJECTIVE

The object of this game is designed for the purpose of Playing Quiz Contest. This system can be used in Schools/Colleges or vice versa for conducting Quiz contest.

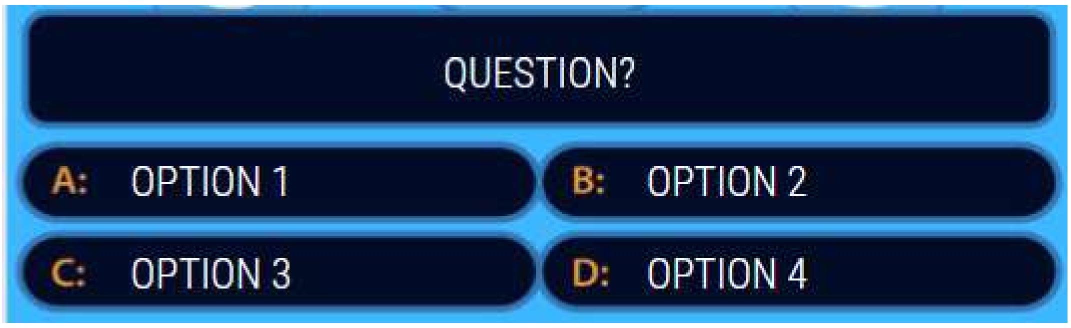


FIG. 1.1 OBJECTIVE DIAGRAM

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE -

- Monitor: Resolution of 1024 x 768 or higher
- RAM: Minimum 2GB or above
- Processor: Intel Core
- Hard Disk: A minimum of 20GB or above of available space
- Keyboard, Mouse

2.2 SOFTWARE-

- Operating System: Ubuntu OS
- Text Editor: G-Edit
- Compiler: GCC

2.3 LANGUAGE-

- C Programming Language

CHAPTER 3

FUNCTIONAL REQUIREMENTS

We have used following functions in this project: -

- game():

  This function asks the users name and displays a instruction message and then asks three questions in Selection Round and goes to ‘test’ if the user answers 2 or more correct answer else it will call menu().

  Goto test label statements asks 10 questions from the user after the selection round and then displays the name and score.

- printMessageCenter():

  This function prints the message in the middle of the head massage. We have passed the message in this function as per the operation.

- headMessage():

  It prints the message on the top of the console and prints the message as per operation.

- Welcome():

  This function displays the first welcome screen of the “Quiz Game project” and asks the user to press any key to access the game application.

- menu():

  This function displays the main menu and asks the user to select the option. If the user selects 0, then the application will close.

- main():

  This function is used to call the function, Welcome() and menu(), to run the program.

CHAPTER 4

FLOWCHART

FIG 4.1 Flowchart for main( )

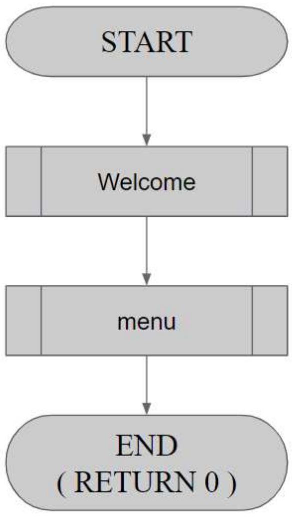
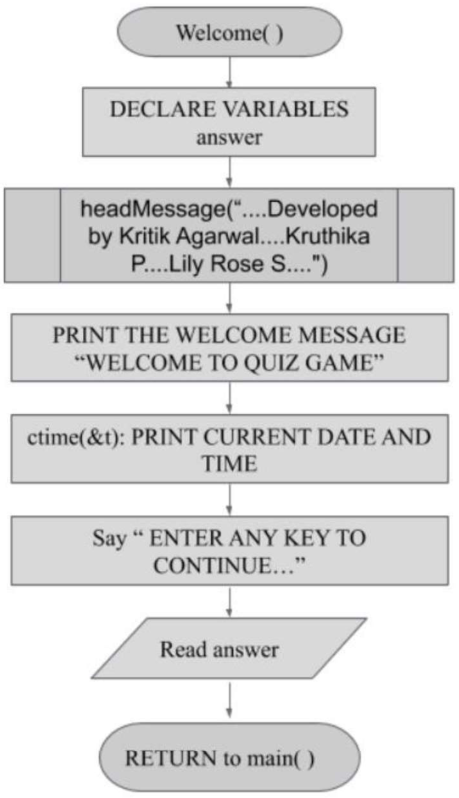


FIG 4.2 Flowchart for Welcome( )


Welcome( )
DECLARE VARIABLES
answer
headMessage("....Developed by Kritik Agarwal....Kruthika P....Lily Rose S....")
PRINT THE WELCOME MESSAGE "WELCOME TO QUIZ GAME"
ctime(&t): PRINT CURRENT DATE AND TIME
Say " ENTER ANY KEY TO CONTINUE…"
Read answer
RETURN to main( )

4.3 Flowchart for menu( )

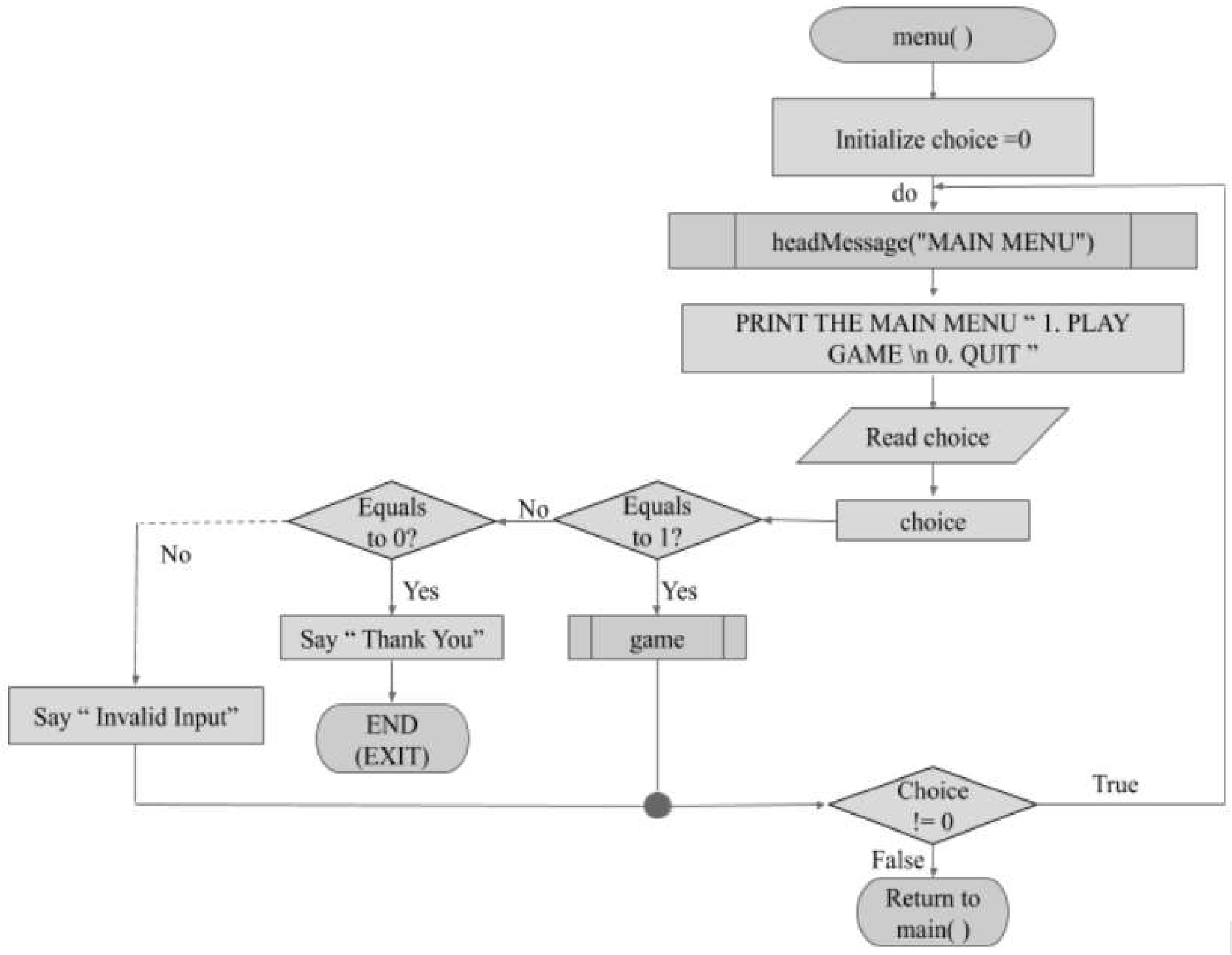


FIG 4.4 Flowchart for headMessage( )

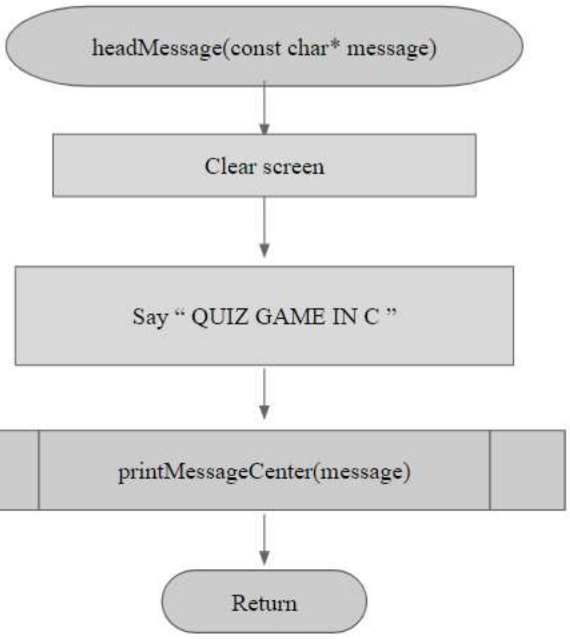


FIG 4.5 Flowchart for printMessageCenter( )

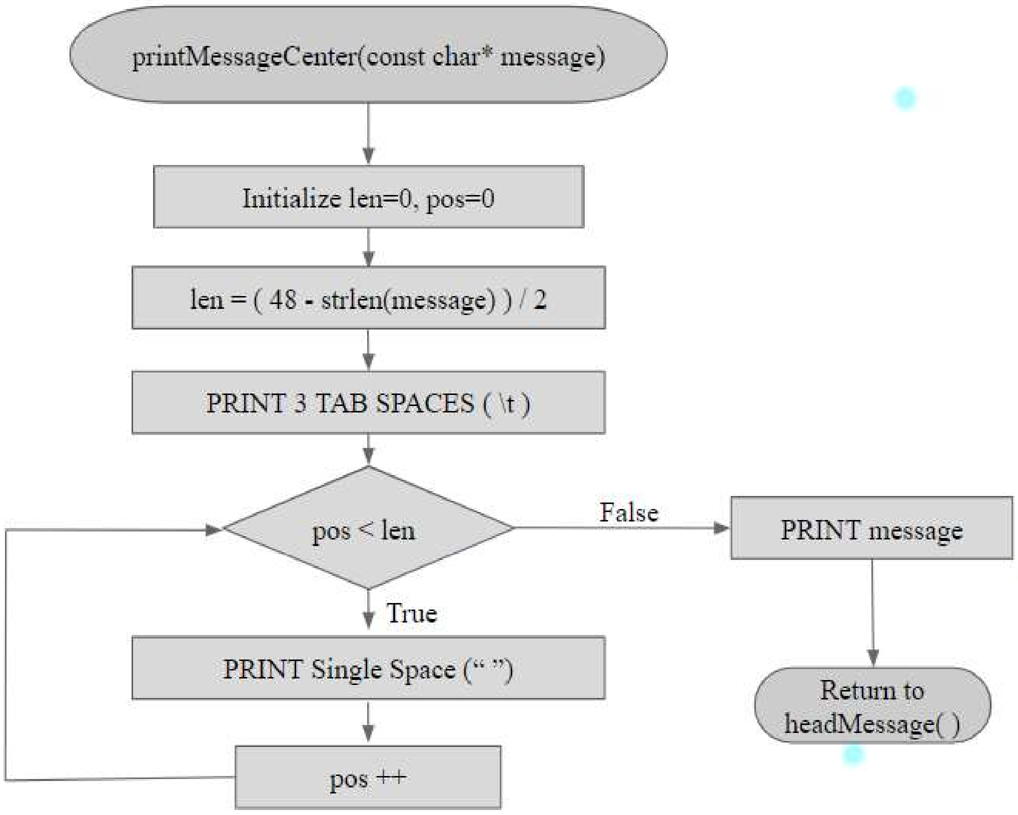
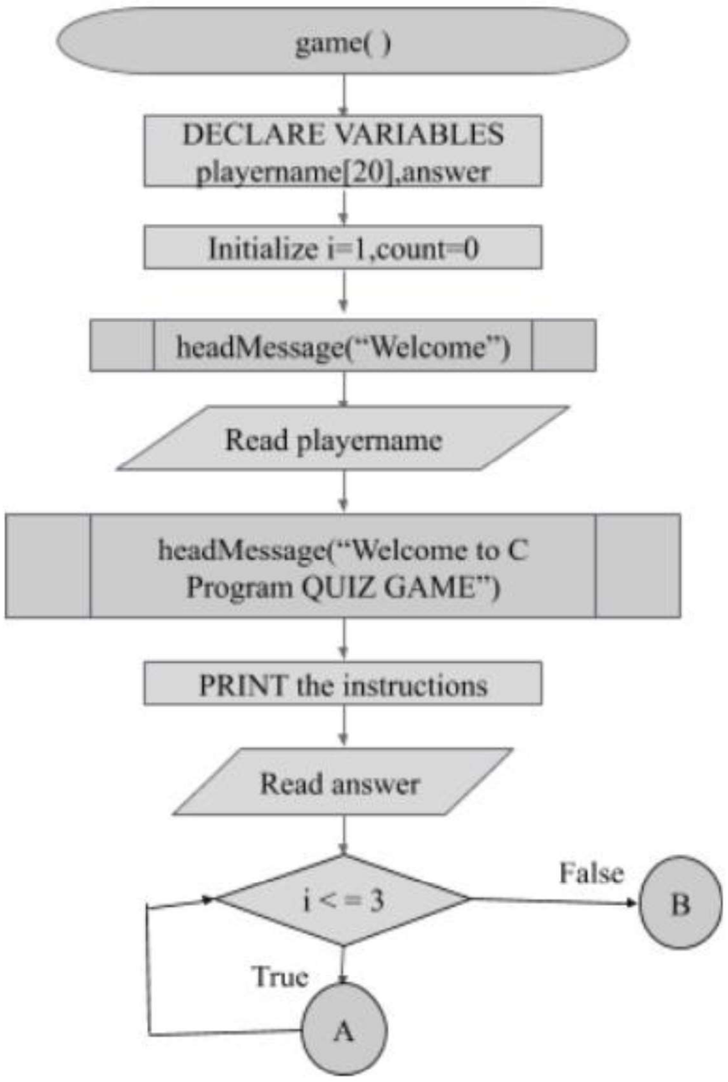
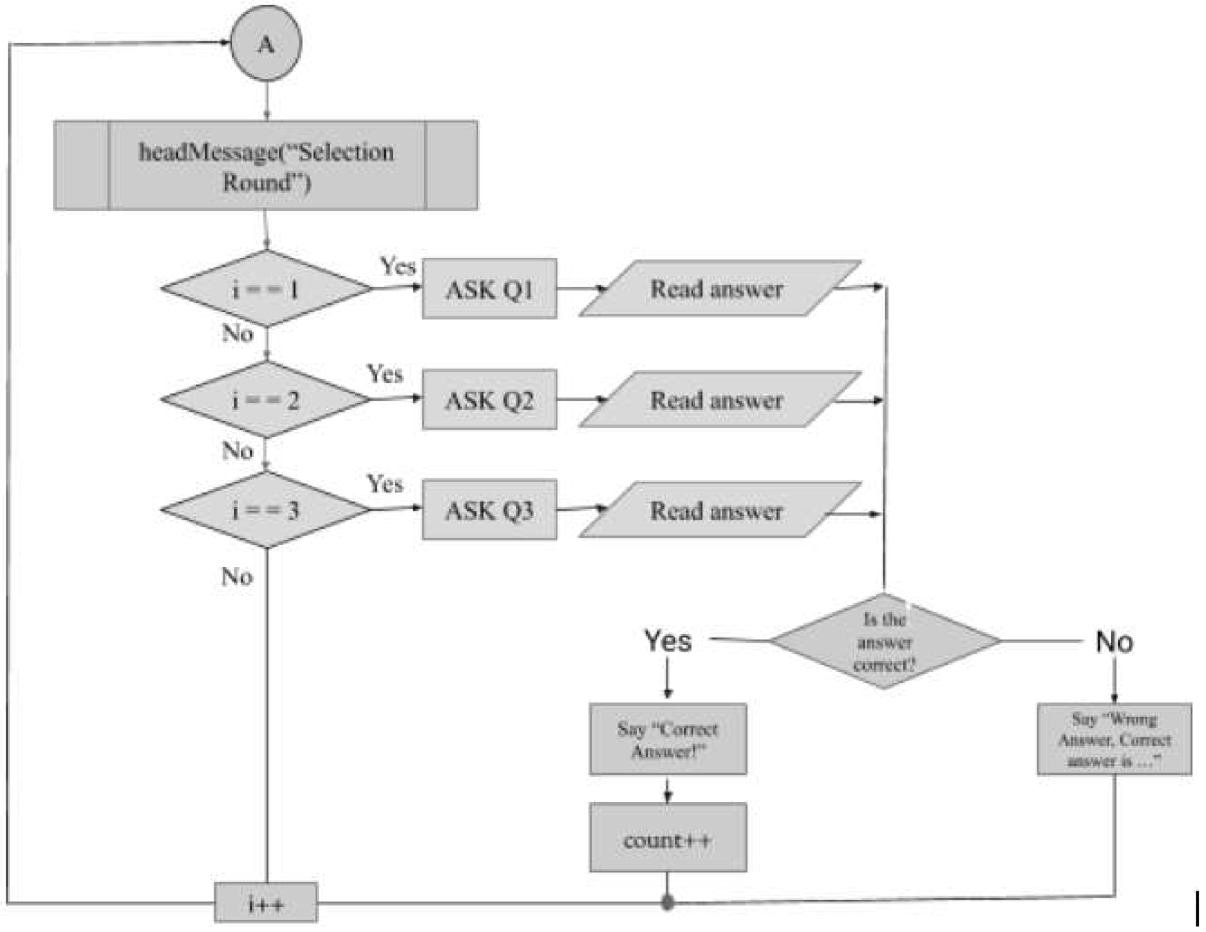


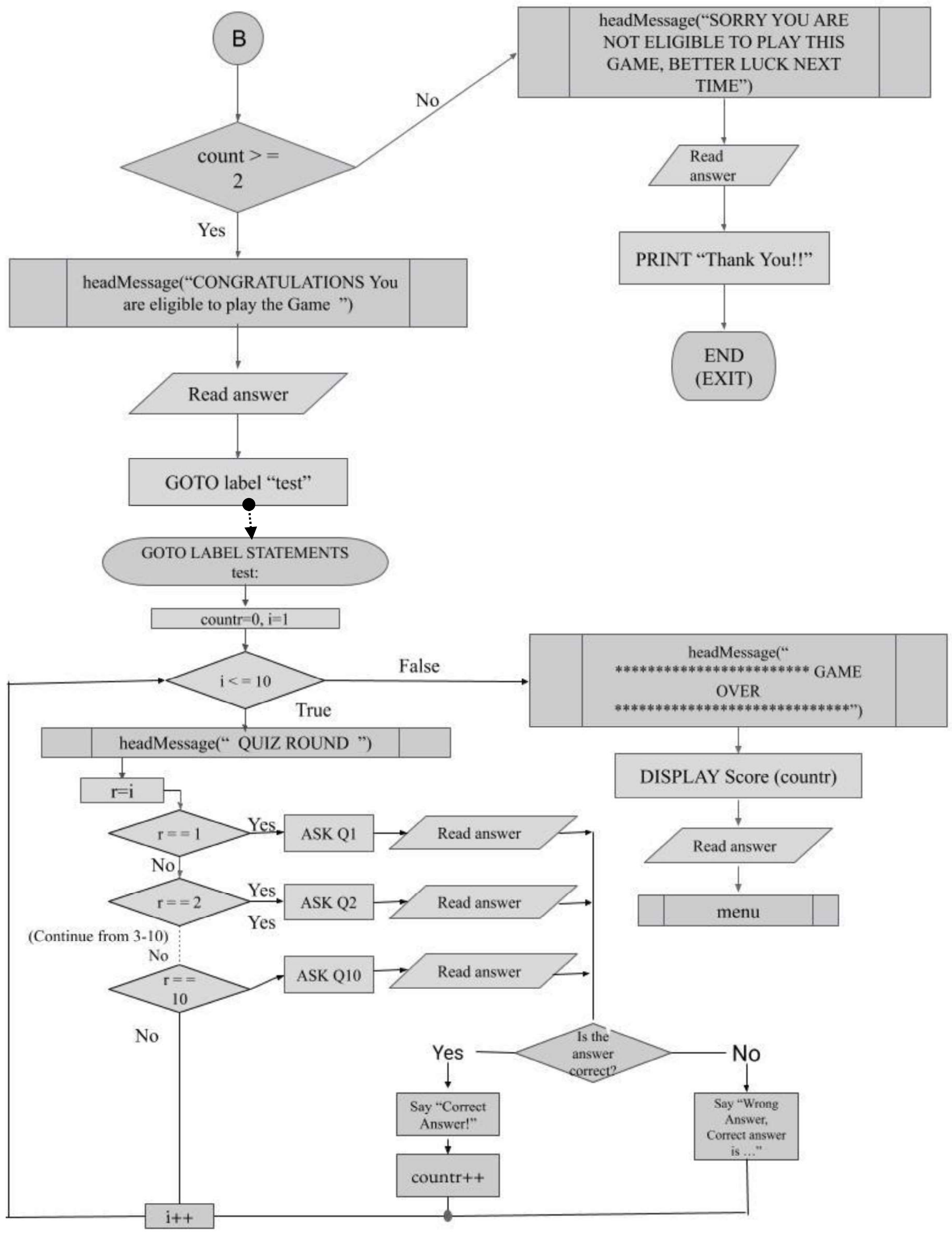
FIG 4.6 Flowchart for game( )



(a)


A
headMessage("Selection Round")
i = = 1
Yes
ASK Q1
Read answer
No
i = = 2
Yes
ASK Q2
Read answer
No
i = = 3
Yes
ASK Q3
Read answer
No
Is the answer correct?
Yes
No
Say "Correct Answer!"
Say "Wrong Answer, Correct answer is ..."
count++
i++

(b)


B
count > = 2
No
Yes
headMessage("SORRY YOU ARE NOT ELIGIBLE TO PLAY THIS GAME, BETTER LUCK NEXT TIME")
Read answer
PRINT "Thank You!!"
END (EXIT)
headMessage("CONGRATULATIONS You are eligible to play the Game ")
Read answer
GOTO label "test"
GOTO LABEL STATEMENTS test:
countr=0, i=1
i < = 10
False
True
headMessage(" ************************ GAME OVER ******************************")
DISPLAY Score (countr)
Read answer
menu
headMessage(" QUIZ ROUND ")
r=i
r = = 1
Yes
ASK Q1
Read answer
No
r = = 2
Yes
ASK Q2
Read answer
Yes
(Continue from 3-10)
No
r = = 10
ASK Q10
Read answer
No
Is the answer correct?
Yes
No
Say "Correct Answer!"
countr++
Say "Wrong Answer, Correct answer is ..."
i++

(c)

CHAPTER 5

ALGORITHM

5.1 main( )

```
Step 1- Start

Step 2- Call Function Welcome( )

Step 3- Call Function menu( )

Step 4- Return 0
```

5.2 Welcome( )

```
Step 1- Start

Step 2- Declare Variables Answer

Step 3- Call Function headMessage("….Developed by Kritik Agarwal…Kruthika
        P….Lily Rose S…. )

Step 4- Print the welcome message "WELCOME TO QUIZ GAME"

Step 5- Print Current Date & Time using ctime(&t)

Step 6- Get input from user

Step 7- Return to main( )
```

5.3 menu( )

```
Step 1-  Start

Step 2-  Initialize choice=0

Step 3-  WHILE choice!=0 do

             Call Function headMessage("MAIN MENU")

             Print the main menu " 1. PLAY GAME \n 0. QUIT "

             Get input from the user "choice"

             Switch choice
```

```
                    If Case 1: Call Function Game( )
                              Break;
                    If Case 2: Print "Thank You"
                              STOP.
                    Else default: Print "Invalid Input"
    Step 4- Stop
```

5.4 headMessage( )

```
    Step 1- Start, get parameter "const char* message"
    Step 2- Clear screen
    Step 3- Print " QUIZ GAME IN C "
    Step 4- Call Function printMessageCenter(message)
    Step 5- Return
```

5.5 printMessageCenter( )

```
    Step 1- Start, get parameter "const char* message"
    Step 2- Initialize len=0, pos=0
    Step 3- Calculate len=(48-strlen(message))/2
    Step 4- Print tab spaces ( \t ) 3 times
    Step 5- For pos < len
                Print Single Space ( "   " )
                Increment pos
    Step 6- Print message
    Step 7- Return to headMessage( )
```

5.6 game( )

```
    Step 1- Start
```

```
Step 2- Declare Variables playername[20], answer

Step 3- Initialize i=1, count=0

Step 4- Call Function headMessage(“Welcome”)

Step 5- Get playername as input

Step 6- Call Function headMessage(“Welcome to C Program QUIZ GAME”)

Step 7- Print the instructions

Step 8- Get input from user

Step 9- WHILE i<=3

            Call Function headMessage(“Selection Round”)

            If i==1 then

                  Ask Q1

                  Get input from the user

                  If Answer==Input then

                        Print “Correct!!”

                        Increment count

                  Else Print “Wrong. Correct Answer is …..”

            Else IF i==2 then

                  Ask Q2

                  Get input from the user

                  If Answer==Input then

                        Print “Correct!!”

                        Increment count

                  Else Print “Wrong. Correct Answer is …..”

            Else IF i==3 then

                  Ask Q3
```

```
                    Get input from the user

                    If Answer==Input then

                              Print “Correct!!”

                              Increment count

                    Else Print “Wrong. Correct Answer is …..”

          Increment i

Step 10- If count > = 2 then

          Call Function headMessage(“CONGRATULATIONS. You are
          eligible to play the Game”)

          Get input from the user

          GOTO test:

     Else

          Call Function headMessage(“SORRY YOU ARE NOT ELIGIBLE TO
          PLAY THIS GAME, BETTER LUCK NEXT TIME”.”)

          Get input from the user

          Print “Thank You”

Step 11- Stop
```

GOTO LABEL “test:”

```
Step 1- Start

Step 2- Declare Variables answer,r

Step 3- Initialize i=1, countr=0

Step 4- WHILE i < = 10

          Call Function headMessage(“YOU ARE IN QUIZ GAME”)

          Initialize r=i

          If r==1 then

                    Ask Q1
```

```
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==2 then
        Ask Q2
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==3 then
        Ask Q3
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==4 then
        Ask Q4
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==5 then
        Ask Q5
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==6 then
        Ask Q6
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==7 then
        Ask Q7
        Get input from the user
        If Answer==Input then
            Print “Correct!!”
            Increment countr
        Else Print “Wrong. Correct Answer is …..”
    Else IF r==8 then
        Ask Q8
        Get input from the user
```

```
                If Answer==Input then
                    Print "Correct!!"
                    Increment countr
                Else Print "Wrong. Correct Answer is ….."
            Else IF r==9 then
                Ask Q9
                Get input from the user
                If Answer==Input then
                    Print "Correct!!"
                    Increment countr
                Else Print "Wrong. Correct Answer is ….."
            Else IF r==10 then
                Ask Q10
                Get input from the user
                If Answer==Input then
                    Print "Correct!!"
                    Increment countr
                Else Print "Wrong. Correct Answer is ….."
            Increment i
    Step 5- Call Function headMessage("*********************** GAME OVER
            *****************************")
    Step 6- Display Score ( Countr )
    Step 7- Get input from user
    Step 8- Call Function menu( )
    Step 9- Return to menu( )
```

CHAPTER 6

PSEUDO CODE

6.1 main( )

```
BEGIN
FUNCTION Welcome( )
FUNCTION menu( )
END
```

6.2 Welcome( )

```
BEGIN
FUNCTION headMessage
WRITE “WELCOME TO QUIZ GAME”
WRITE Current Date and Time
READ answer
RETURN
END
```

6.3 menu( )

```
BEGIN
choice=0
WHILE choice!=0 DO
    FUNCTION headMessage
    WRITE “ 1. PLAY GAME \n 0. QUIT ”
    READ choice
    IF choice==1 THEN
        FUNCTION game( )
```

```
                    BREAK
            ELSE IF choice==2 THEN
                    WRITE Thank You
                    EXIT
            ELSE
                    WRITE Invalid Input
            END IF
    END
```

6.4 headMessage( )

```
    BEGIN
    INITIALIZE PARAMETER message
    CLEAR SCREEN
    WRITE QUIZ GAME IN C
    FUNCTION printMessageCenter
    RETURN
    END
```

6.5 printMessageCenter( )

```
    BEGIN
    INITIALIZE PARAMETER message
    len=0, pos=0
    CALCULATE len=(48-strlen(message))/2
    WRITE TAB SPACE(\t) 3 TIMES
    WHILE pos<len
                    WRITE SINGLE SPACES  ( "   " )
```

```
            CALCULATE pos++
    WRITE message
    RETURN
    END
```

6.6 game( )

```
    BEGIN
    DECLARE VARIABLES playername[20], answer
    i=1, count=0
    FUNCTION headMessage
    READ playername
    FUNCTION headMessage
    WRITE THE INSTRUCTIONS
    READ answer
    WHILE i<=3
            FUNCTION headMessage
            IF i==1 THEN
                WRITE Q1
                READ answer
                IF answer==input THEN
                    WRITE "Correct!!"
                    CALCULATE count++
                ELSE
                WRITE "Wrong. Correct Answer is ….."
            ELSE IF i==2 THEN
                WRITE Q2
```

```
                    READ answer
                    IF answer==input THEN
                         WRITE "Correct!!"
                         CALCULATE count++
                    ELSE
                    WRITE "Wrong. Correct Answer is ….."
               Else IF i==3 THEN
                    WRITE Q1
                    READ answer
                    IF answer==input THEN
                         WRITE "Correct!!"
                         CALCULATE count++
                    ELSE
                    WRITE "Wrong. Correct Answer is ….."
               END IF
               CALCULATE i++
     IF count > = 2 THEN
               FUNCTION headMessage
               READ answer
               GOTO test
     ELSE
          FUNCTION headMessage
          READ answer
          WRITE "Thank You"
     END IF
```

```
    END
```

GOTO LABEL “test:”

```
    BEGIN
    i=1, countr=0
    WHILE i < = 10
            FUNCTION headMessage
            r=i
            IF r==1 THEN
                WRITE Q1
                READ answer
                IF answer==input THEN
                    WRITE “Correct!!”
                    CALCULATE countr++
                ELSE
                WRITE “Wrong. Correct Answer is …..”
            ELSE IF r==2 THEN
                WRITE Q1
                READ answer
                IF answer==input THEN
                    WRITE “Correct!!”
                    CALCULATE countr++
                ELSE
                WRITE “Wrong. Correct Answer is …..”
            ELSE IF r==3 THEN
                WRITE Q3
```

```
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==4 THEN
        WRITE Q4
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==5 THEN
        WRITE Q5
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==6 THEN
        WRITE Q6
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==7 THEN
        WRITE Q7
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==8 THEN
        WRITE Q8
        READ answer
        IF answer==input THEN
            WRITE “Correct!!”
            CALCULATE countr++
        ELSE
        WRITE “Wrong. Correct Answer is …..”
    ELSE IF r==9 THEN
        WRITE Q9
        READ answer
        IF answer==input THEN
```

```
                    WRITE “Correct!!”
                    CALCULATE countr++
                ELSE
                WRITE “Wrong. Correct Answer is …..”
            ELSE IF r==10 THEN
                WRITE Q10
                READ answer
                IF answer==input THEN
                    WRITE “Correct!!”
                    CALCULATE countr++
                ELSE
                WRITE “Wrong. Correct Answer is …..”
            END IF
            CALCULATE i++
    FUNCTION headMessage
    WRITE Score ( Countr )
    READ answer
    FUNCTION menu
    RETURN
    END
```

CHAPTER 7

IMPLEMENTATION

```
//QUIZ GAME DEVELOPED BY KRITIK AGARWAL, KRUTHIKA P, LILY ROSE S

// Required Header Files

#include<time.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>


//Function Declaration

void printMessageCenter();

void headMessage();

void Welcome();

void game();

void menu();


//Program Starts here

int main()

{

        Welcome();

        menu();

        return 0;

}
```

```
//Function 1

void printMessageCenter(const char* message)

{

   int len =0;

   int pos = 0;

   //calculate how many space need to print

   len = (48 - strlen(message))/2;

   printf("\t\t\t");

   for(pos =0 ; pos < len ; pos++)

               printf(" ");

   printf("%s",message);

}

//Function 2

void headMessage(const char *message)

{

   system("clear");

printf("\t\t#########################################################################
#######");

   printf("\n\t\t#############                QUIZ GAME in C              #############");

printf("\n\t\t#######################################################################
#########");

   printf("\n\t\t------------------------------------------------------------------------\n");

   printMessageCenter(message);

   printf("\n\t\t------------------------------------------------------------------------");

}
```

```
//Function 3

void Welcome()

{

    char answer;

    time_t t;

    time(&t);

    headMessage("....Developed by Kritik Agarwal....Kruthika P....Lily Rose S....");

    printf("\n\n\n\n\n");

    printf("\n\t\t\t  **-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**\n");

    printf("\n\t\t\t        =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");

    printf("\n\t\t\t        =                 WELCOME                   =");

    printf("\n\t\t\t        =                    TO                     =");

    printf("\n\t\t\t        =                  QUIZ                     =");

    printf("\n\t\t\t        =                  GAME                     =");

    printf("\n\t\t\t        =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");

    printf("\n\t\t\t  **-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**\n");

    printf("\n\t\t\t  Current date and time : %s                    ",ctime(&t));

    printf("\n\t\t\t  **-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**-**\n");

    printf("\n\n\n\t\t\t Enter any key to continue.....");

    scanf(" %c",&answer);

}
```

```
//Function 4

void game()

{

        char playername[20],answer;

        int count=0,i=1,countr=0,r;

        headMessage("Welcome");

        printf("\n\n\n\n\n\n\n\n\n\n\t\tEnter your name:");

        scanf("%s",playername);

        headMessage("Welcome to C Program Quiz Game");

        printf("\n\n\t\t Here are some tips you might wanna know before playing:");

        printf("\n\t\t ----------------------------------------------------------------------");

        printf("\n\t\t >> There are 2 rounds in this Quiz Game,WARMUP ROUND & CHALLENGE ROUND");

        printf("\n\t\t >> In warmup round you will be asked total 3 questions to test your");

        printf("\n\t\t     general knowledge. You are eligible to play the game if you give atleast 2");

        printf("\n\t\t     right answers, otherwise you can't proceed further to the Challenge Round.");

        printf("\n\t\t >> Your game starts with CHALLENGE ROUND. In this round you will be asked a");

        printf("\n\t\t    total of 10 questions. Each right answer will be awarded 1 point");

        printf("\n\t\t >> You will be given 4 options and you have to press A, B ,C or D for the");

        printf("\n\t\t    right option.");

        printf("\n\t\t >> You will be asked questions continuously, till right answers are given");

        printf("\n\t\t >> No negative marking for wrong answers!");

        printf("\n\n\t\t\t!!!!!!!!!!!!! ALL THE BEST !!!!!!!!!!!!");
        printf("\n\n\t\tPress any key to start the game!");

        scanf(" %c",&answer);

        while(i<=3)

        {

                headMessage("Selection Round");

                if(i==1)

                {

                        printf("\n\n\t\t1. Which is valid C expression?");

                        printf("\n\n\t\tA)           int           my_num=100,000;\t\tB)           int
my_num=100000;\n\n\t\tC) int my num=1000;\t\tD) int $my_num=10000;");

                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");

                        scanf(" %c",&answer);

                        if(answer=='B'||answer=='b')

                        {

                                printf("\n\n\t\tCorrect!!!\n");

                                count++;

                        }

                        else

                        printf("\n\n\t\tWrong!!! The correct answer is B.\n");

                }

        else if(i==2)

                {

                        printf("\n\n\t\t2. Which among the following is NOT a logical or
relational operator?");

                        printf("\n\n\t\tA) !=\t\tB) ==\n\n\t\tC) ||\t\tD) =");

                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                        scanf(" %c",&answer);

                        if(answer=='D'||answer=='d')

                        {

                                printf("\n\n\t\tCorrect!!!\n");

                                count++;

                        }

                                else

                                printf("\n\n\t\tWrong!!! The correct answer is D.\n");


                }

                else if(i==3)

                {

                        printf("\n\n\t\t3. Which animal laughs like human being?");

                   printf("\n\n\t\tA.Polar Bear\t\tB.Hyena\n\n\t\tC.Donkey\t\tD.Chimpanzee");

                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");

                        scanf(" %c",&answer);

                        if(answer=='B'||answer=='b')

                        {

                                printf("\n\n\t\tCorrect!!!\n");

                                count++;

                        }

                        else

                                printf("\n\n\t\tWrong!!! The correct answer is B.Hyena\n");

                }

                i++;
        }//Loop Ended

        if(count>=2)

        {

                headMessage("CONGRATULATIONS. You are eligible to play the Game.");

                printf("\n\n\n\n\t\t!Press any key to Start the Game!: ");

                scanf(" %c",&answer);

                goto test;

        }

        else

        {

                headMessage("SORRY YOU ARE NOT ELIGIBLE TO PLAY THIS GAME,
                BETTER LUCK NEXT TIME");

                printf("\n\n\t\t\tEnter 'Q' to exit.\t");

                scanf(" %c",&answer);

                if(answer=='Q'||answer=='q')

                {

                        printf("\n\n\n\t\t\t\tThank you!!!\n\n\n\n\n");

                        exit(1);

                }

        }

//GOTO LABEL DECLARATION

        test:

                countr=0;

                for(i=1;i<=10;i++)

                {
```

```
                headMessage(" QUIZ ROUND ");
                r=i;
                switch(r)
                {
                        case 1:
                        printf("\n\n\t\t1. What is the National Game of England?");
        printf("\n\n\t\tA.Football\t\tB.Basketball\n\n\t\tC.Cricket\t\tD.Baseball");
                                printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                                scanf(" %c",&answer);
                                if(answer=='C'||answer=='c')
                                {
                                        printf("\n\n\t\tCorrect!!!\n");
                                        countr++;
                                        break;
                                }
                                else
                                {
                        printf("\n\n\t\tWrong!!! The correct answer is C.Cricket\n");
                                        break;
                                }
                        case 2:
                                printf("\n\n\t\t2. Study of Earthquake is called...........");
        printf("\n\n\t\tA.Seismology\t\tB.Cosmology\n\n\t\tC.Orology\t\tD.Etimology");
                                printf("\n\n\t\t\tENTER YOUR CHOICE: ");
```

```
                                        scanf(" %c",&answer);
                                        if(answer=='A'|| answer=='a')
                                        {
                                                printf("\n\n\t\tCorrect!!!\n");
                                                countr++;
                                                break;
                                        }
                                        else
                                        {
                                printf("\n\nWrong!!! The correct answer is A.Seismology");
                                                break;
                                        }
                        case 3:
printf("\n\n\t\t3. Among the top 10 highest peaks in the world, how many lie in Nepal? ");
                                        printf("\n\n\t\tA.6\t\tB.7\n\n\t\tC.8\t\tD.9");
                                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                                        scanf(" %c",&answer);
                                        if(answer=='C'||answer=='c')
                                        {
                                                printf("\n\n\t\tCorrect!!!\n");
                                                countr++;
                                                break;
                                        }
                                        else
                                        {
```

```
                                        printf("\n\n\t\tWrong!!! The correct answer is C.8\n");
                                                    break;
                                          }
                              case 4:
                                          printf("\n\n\t\t4. The Laws of Electromagnetic Induction
were given by?");

        printf("\n\n\t\tA.Faraday\t\tB.Tesla\n\n\t\tC.Maxwell\t\tD.Coulomb");
                                          printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                                          scanf(" %c",&answer);
                                          if(answer=='A'|| answer=='a')
                                          {
                                                    printf("\n\n\t\tCorrect!!!\n");
                                                    countr++;
                                                    break;
                                          }
                                          else
                                          {
                                                    printf("\n\n\t\tWrong!!! The correct answer is
A.Faraday\n");
                                                    break;
                                          }
                              case 5:
                                          printf("\n\n\t\t5. In what unit is electric power
measured?");

        printf("\n\n\t\tA.Coulomb\t\tB.Watt\n\n\t\tC.Power\t\tD.Units");
```

```
                    printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                    scanf(" %c",&answer);
                    if(answer=='B'||answer=='b')
                    {
                        printf("\n\n\t\tCorrect!!!\n");
                        countr++;
                        break;
                    }
                    else
                    {
                    printf("\n\n\t\tWrong!!! The correct answer is
B.Power\n");
                        break;
                    }
                case 6:
                    printf("\n\n\t\t6. Which element is found in Vitamin
B12?");

    printf("\n\n\t\tA.Zinc\t\tB.Cobalt\n\n\t\tC.Calcium\t\tD.Iron");
                    printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                    scanf(" %c",&answer);
                    if(answer=='B'||answer=='b')
                    {
                        printf("\n\n\t\tCorrect!!!\n");
                        countr++;
                        break;
```

```
                                        }
                                        else
                                        {
                                              printf("\n\n\t\tWrong!!! The correct answer is
B.Cobalt\n");
                                              break;
                                        }

                          case 7:
                                        printf("\n\n\t\t7. What is the National Name of Japan?");
      printf("\n\n\t\tA.Polska\t\tB.Hellas\n\n\t\tC.Drukyul\t\tD.Nippon");
                                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                                        scanf(" %c",&answer);
                                        if(answer=='D'||answer=='d')
                                        {
                                              printf("\n\n\t\tCorrect!!!\n");
                                              countr++;
                                              break;
                                        }
                                        else
                                        {
                                              printf("\n\n\t\tWrong!!! The correct answer is
D.Nippon\n");
                                              break;
                                        }
```

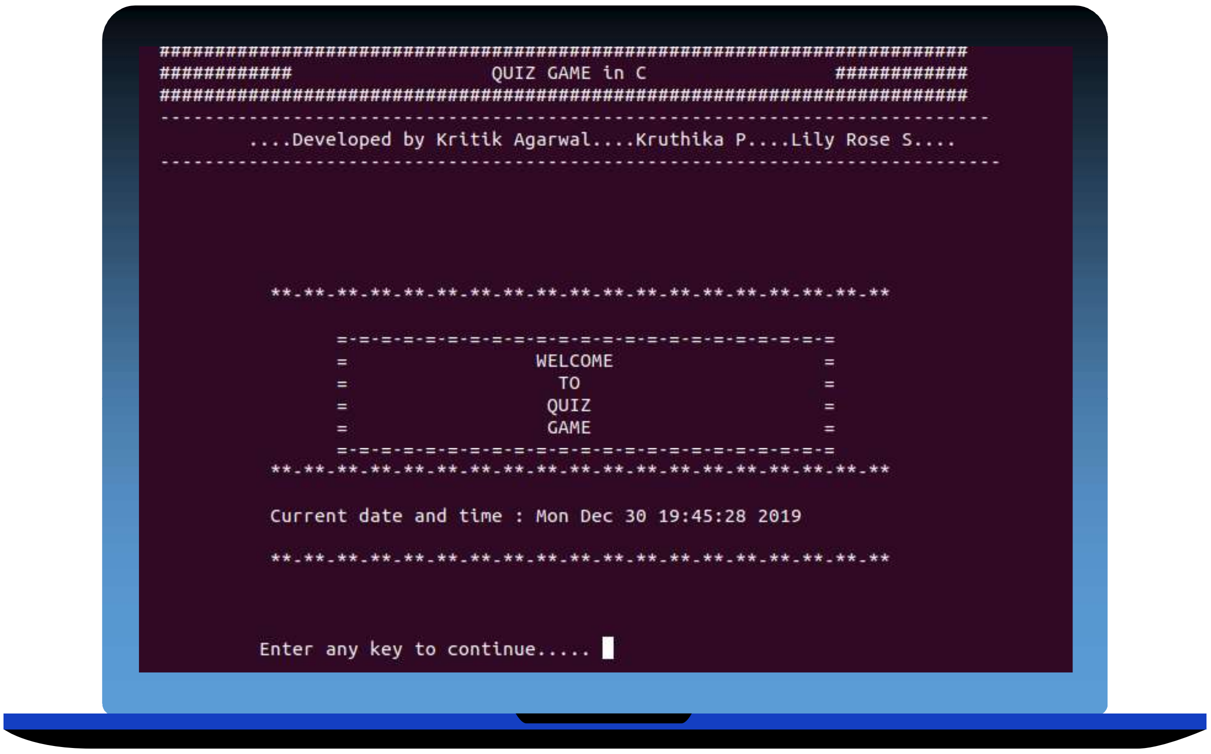
```
                case 8:
                        printf("\n\n\t\t8. How many times a piece of paper can be folded at the most?");
                        printf("\n\n\t\tA.6\t\tB.7\n\n\t\tC.8\t\tD.Depends on the size of paper");
                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                        scanf(" %c",&answer);
                        if(answer=='B'|| answer=='b')
                        {
                                printf("\n\n\t\tCorrect!!!\n");
                                countr++;
                                break;
                        }
                        else
                        {
                                printf("\n\n\t\tWrong!!! The correct answer is B.7\n");
                                break;
                        }
                case 9:
                        printf("\n\n\t\t9. What is the capital of Denmark?");
    printf("\n\n\t\tA.Copenhagen\t\tB.Helsinki\n\n\t\tC.Ajax\t\tD.Galatasaray");
                        printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                        scanf(" %c",&answer);
                        if(answer=='A'|| answer=='a')
                        {
```

```
                        printf("\n\n\t\tCorrect!!!\n");
                        countr++;
                        break;
                    }
                    else
                    {
                        printf("\n\n\t\tWrong!!! The correct answer is
A.Copenhagen\n");
                        break;
                    }
            case 10:
                    printf("\n\n\t\t10. Which is the longest River in the
world?");

    printf("\n\n\t\tA.Nile\t\tB.Koshi\n\n\t\tC.Ganga\t\tD.Amazon");
                    printf("\n\n\t\t\tENTER YOUR CHOICE: ");
                    scanf(" %c",&answer);
                    if(answer=='A'|| answer=='a')
                    {
                        printf("\n\n\t\tCorrect!!!\n");
                        countr++;
                        break;
                    }
                    else
                    {
                        printf("\n\n\t\tWrong!!! The correct answer is
A.Nile\n");
```

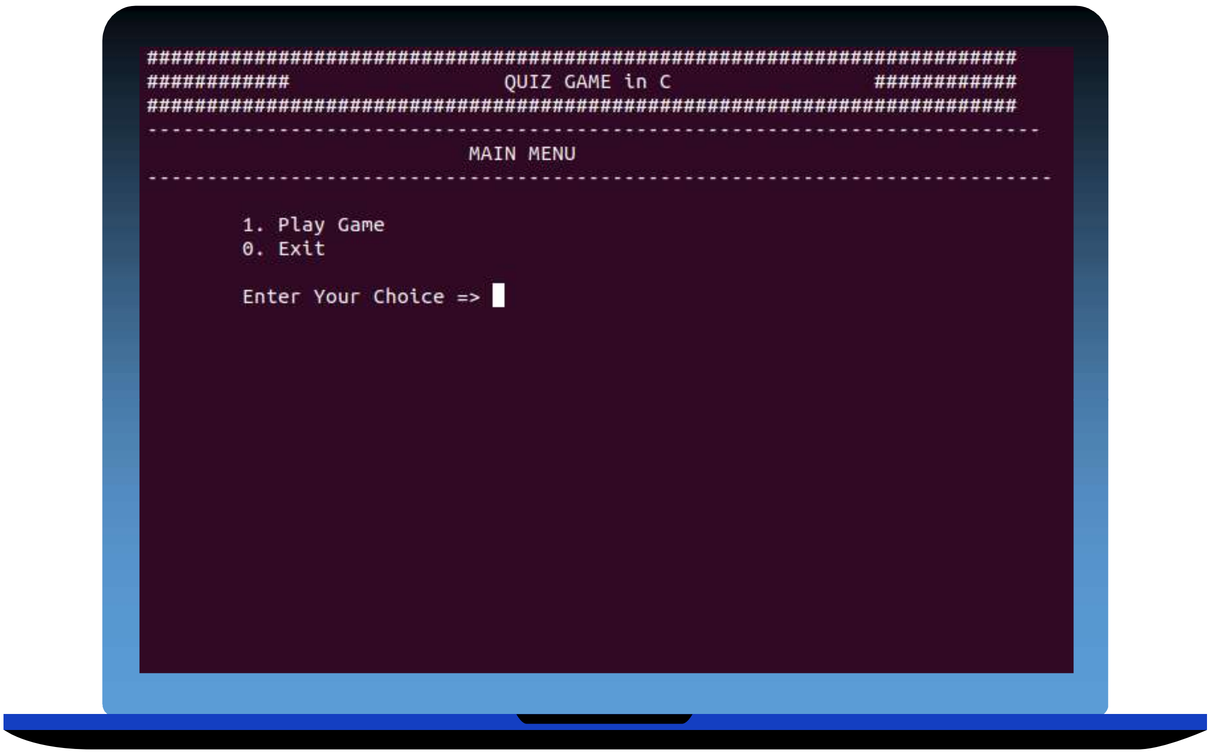
```
                                        break;
                                }
                        }
                }
                headMessage("********************** GAME OVER
****************************");
                printf("\n\t\tCONGRATULATIONS %s You scored %d",playername,countr);
                printf("\n\n\t\tEnter 'M' for main menu.\n");
                scanf(" %c",&answer);
                if(answer=='M' || answer=='m')
                {
                        printf("\n\n\n\t\t\tThank you!!!\n\n\n\n\n");
                        menu();
                }
}
//Function 5
void menu()
{
    int choice = 0;
    do
    {
        headMessage("MAIN MENU");
        printf("\n\n\t\t\t1. Play Game");
        printf("\n\t\t\t0. Exit");
        printf("\n\n\t\t\tEnter Your Choice => ");
        scanf("%d",&choice);

        switch(choice)

        {

           case 1:

                        game();

                        break;

           case 0:

                        printf("\n\n\n\t\t\tThank you!!!\n\n\n\n\n");

                        exit(1);

                        break;

           default:

                        printf("\n\n\n\t\t\tINVALID INPUT!!! Try again...");

        }//Switch Ended

    }while(choice!=0);//Loop Ended

}
```

CHAPTER 8

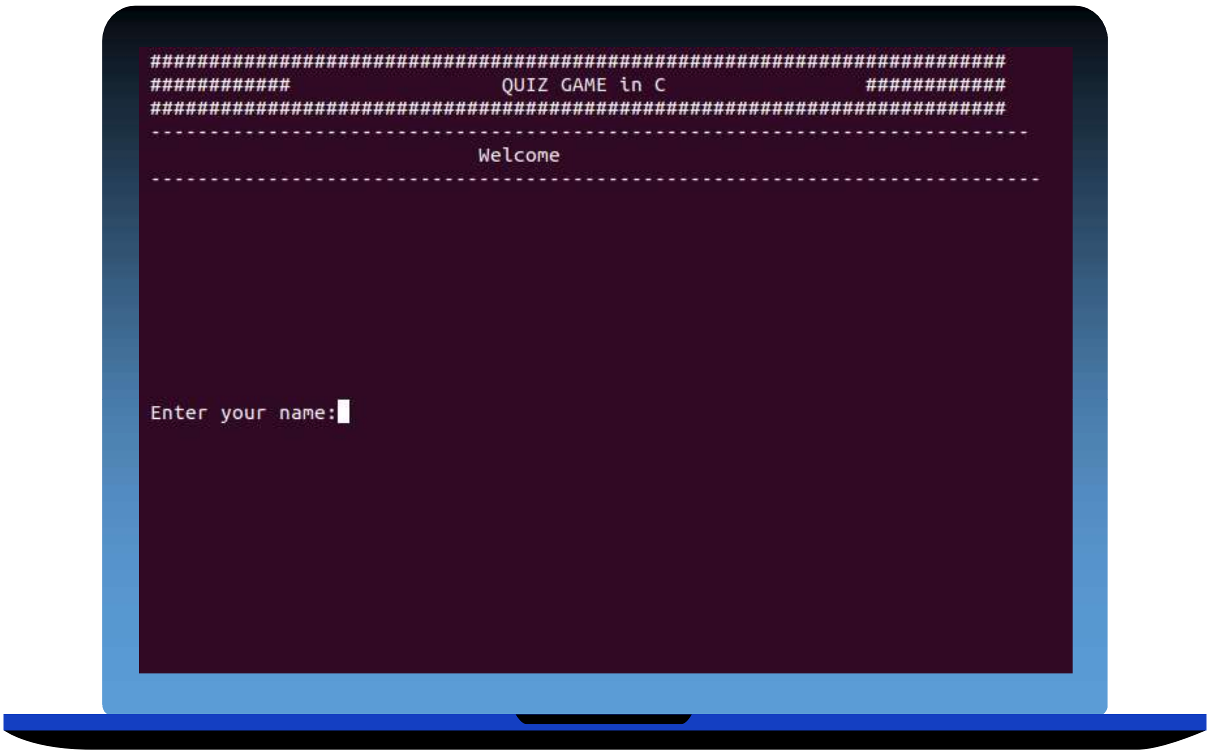
RESULTS



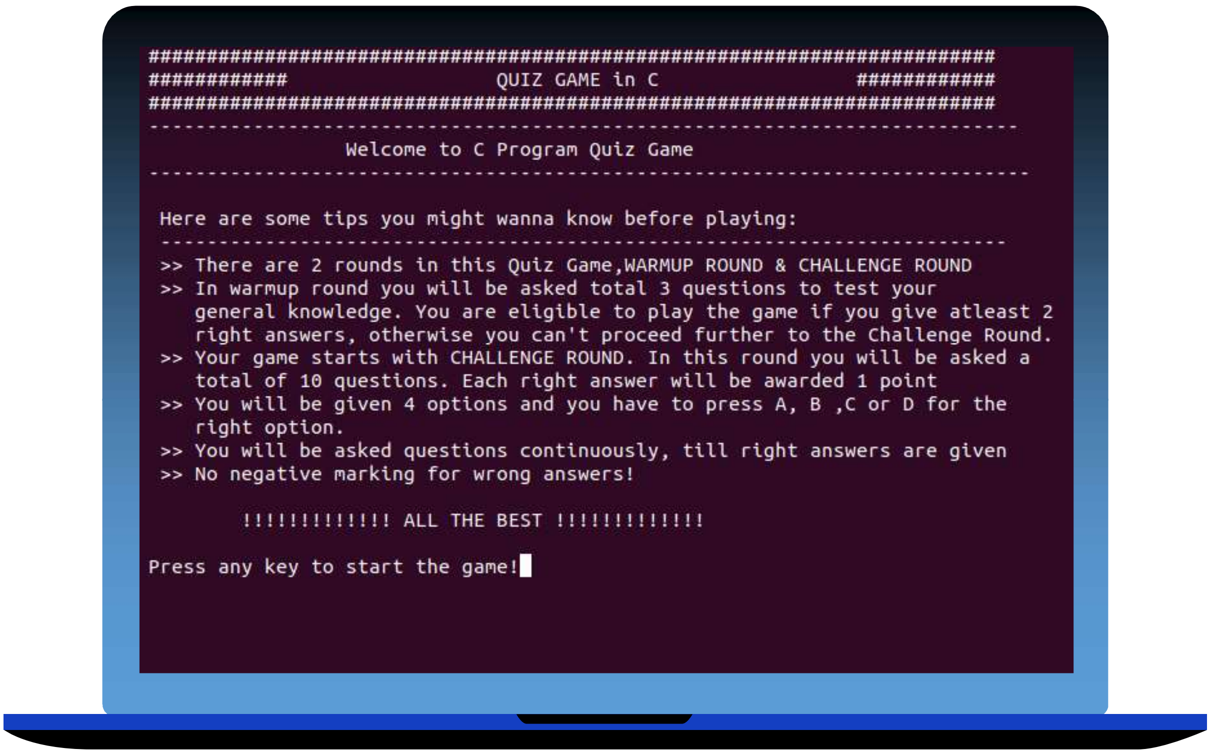
```
##########################################################################
############                 QUIZ GAME in C                  ############
##########################################################################
----------------------------------------------------------------------------
        ....Developed by Kritik Agarwal....Kruthika P....Lily Rose S....
-----------------------------------------------------------------------------




          **.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**

                =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
                =                 WELCOME                     =
                =                   TO                        =
                =                  QUIZ                       =
                =                  GAME                       =
                =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
          **.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**

          Current date and time : Mon Dec 30 19:45:28 2019

          **.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**.**


         Enter any key to continue.....
```



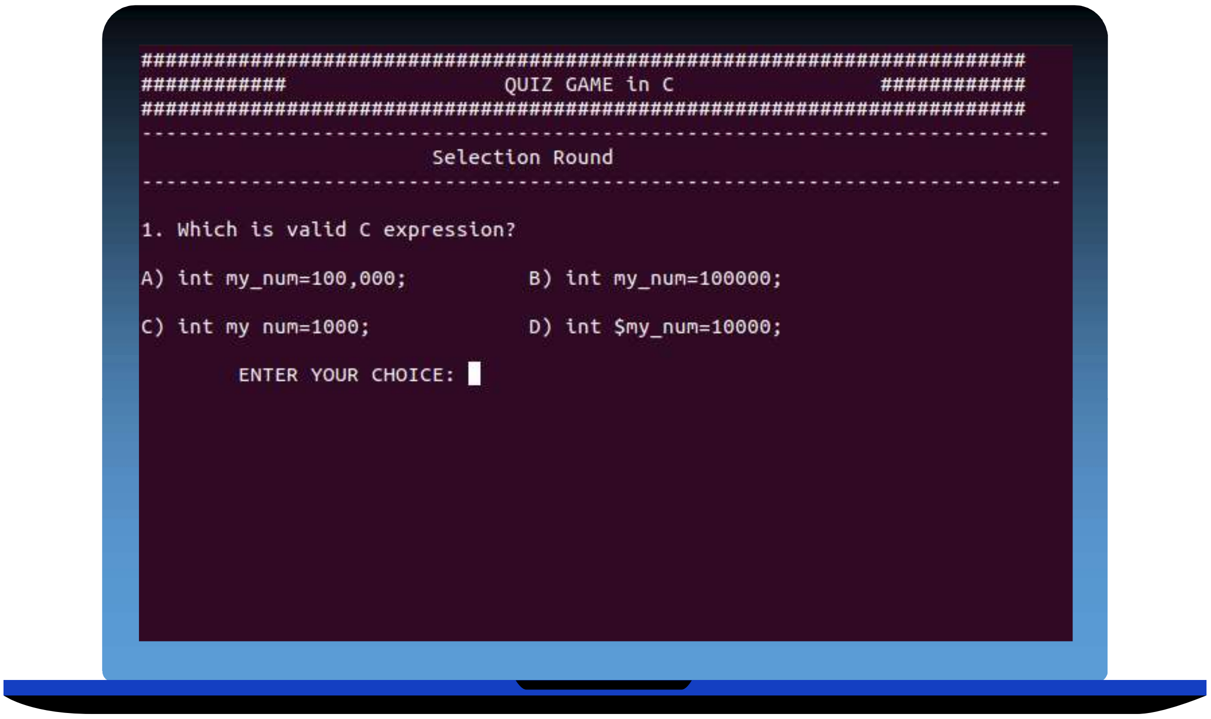
```
#########################################################################
############                 QUIZ GAME in C                  ############
#########################################################################
---------------------------------------------------------------------------
                           MAIN MENU
----------------------------------------------------------------------------

        1. Play Game
        0. Exit

        Enter Your Choice =>
```



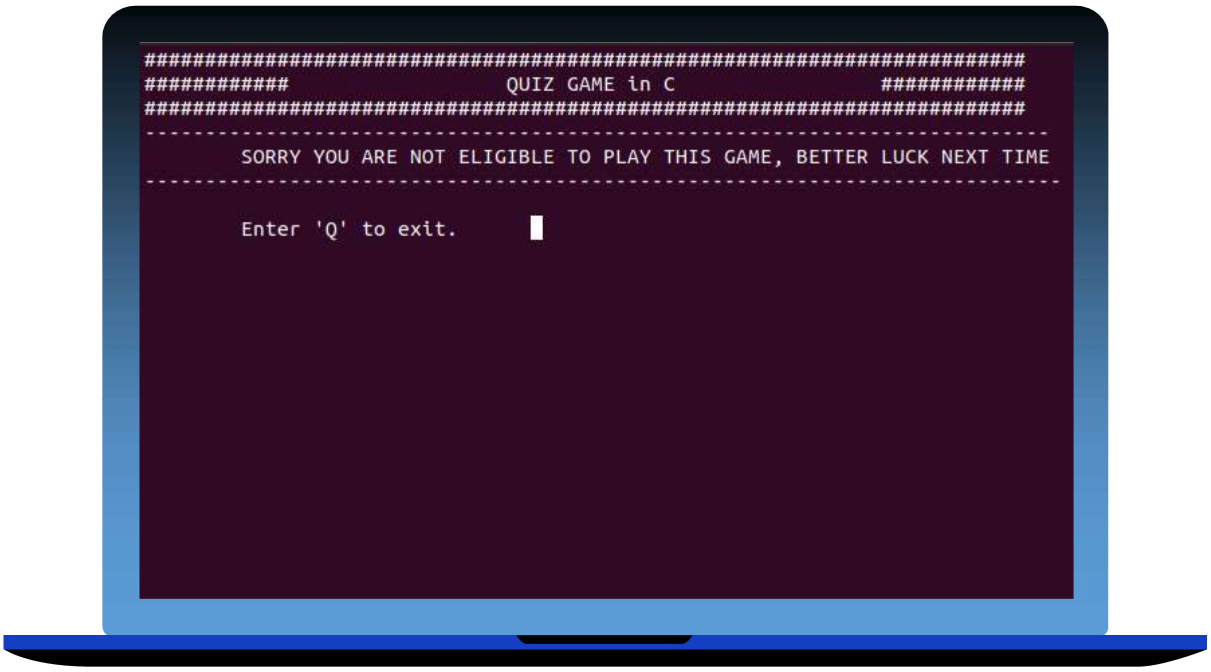
```
#########################################################################
############                 QUIZ GAME in C                  ############
#########################################################################
---------------------------------------------------------------------------
                             Welcome
---------------------------------------------------------------------------







Enter your name:
```



```
########################################################################
############                 QUIZ GAME in C                ############
########################################################################
---------------------------------------------------------------------------
                 Welcome to C Program Quiz Game
----------------------------------------------------------------------------

 Here are some tips you might wanna know before playing:
 -------------------------------------------------------------------------
 >> There are 2 rounds in this Quiz Game,WARMUP ROUND & CHALLENGE ROUND
 >> In warmup round you will be asked total 3 questions to test your
    general knowledge. You are eligible to play the game if you give atleast 2
    right answers, otherwise you can't proceed further to the Challenge Round.
 >> Your game starts with CHALLENGE ROUND. In this round you will be asked a
    total of 10 questions. Each right answer will be awarded 1 point
 >> You will be given 4 options and you have to press A, B ,C or D for the
    right option.
 >> You will be asked questions continuously, till right answers are given
 >> No negative marking for wrong answers!

        !!!!!!!!!!!!! ALL THE BEST !!!!!!!!!!!!!

Press any key to start the game!
```

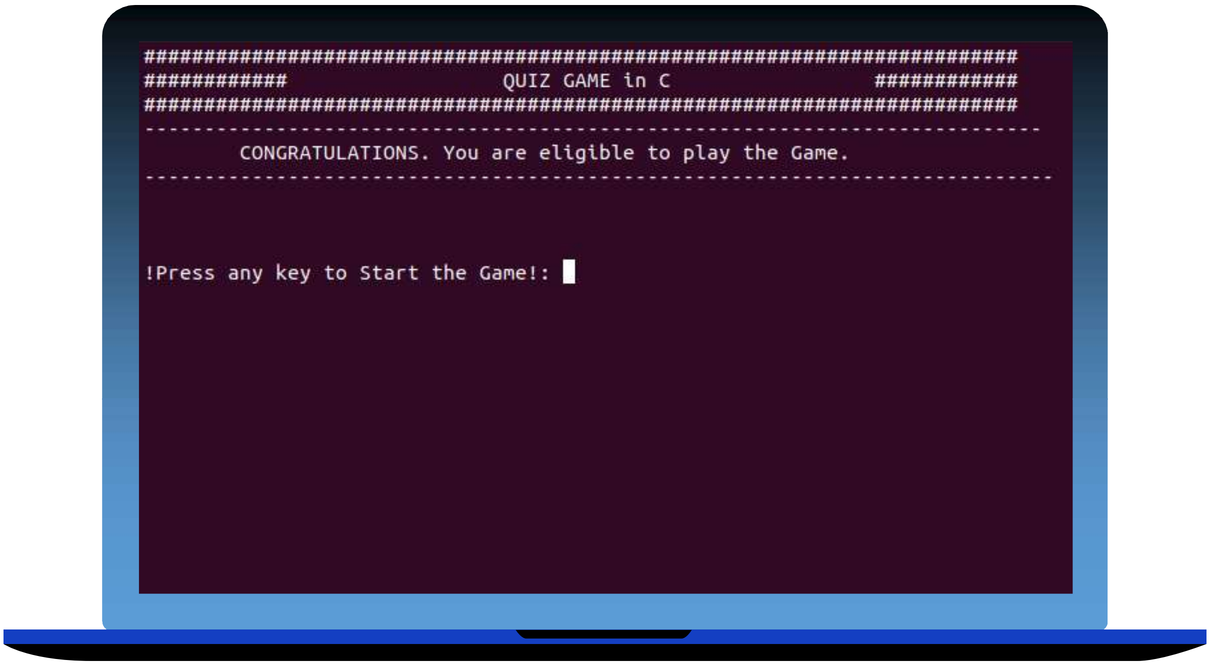


```
############################################################################
############                  QUIZ GAME in C                 ############
############################################################################
-----------------------------------------------------------------------------
                         Selection Round
-----------------------------------------------------------------------------

1. Which is valid C expression?

A) int my_num=100,000;          B) int my_num=100000;

C) int my num=1000;             D) int $my_num=10000;

        ENTER YOUR CHOICE:
```

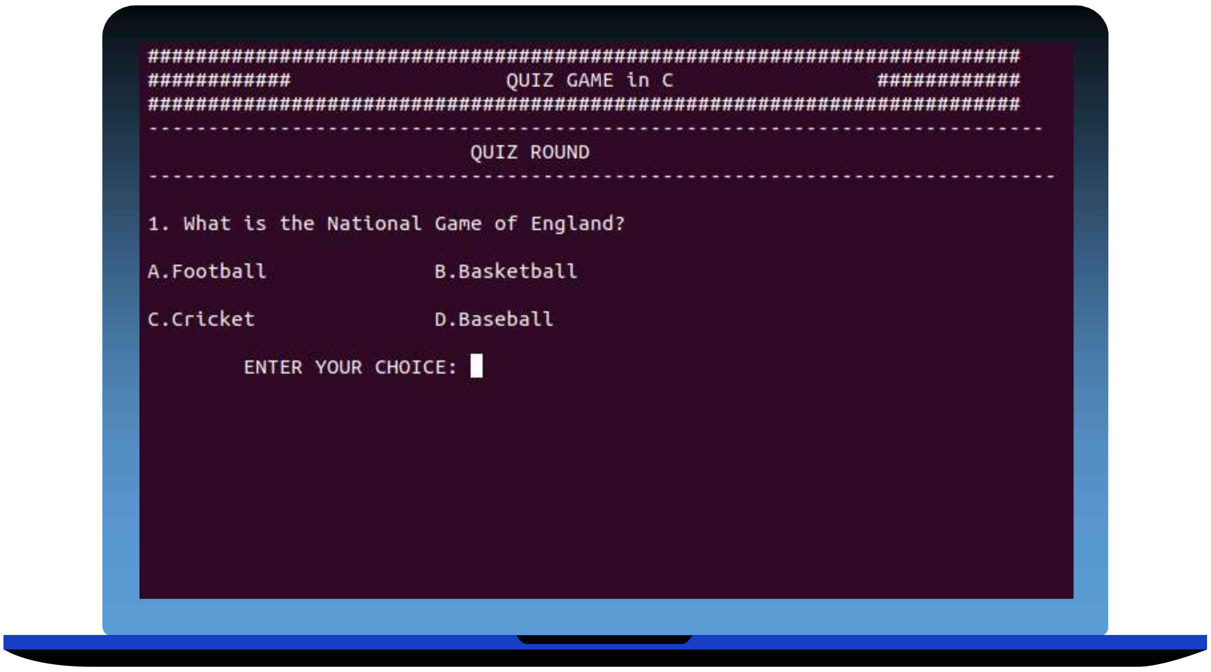


```
##########################################################################
############                 QUIZ GAME in C                 ############
##########################################################################
----------------------------------------------------------------------------
        SORRY YOU ARE NOT ELIGIBLE TO PLAY THIS GAME, BETTER LUCK NEXT TIME
----------------------------------------------------------------------------

        Enter 'Q' to exit.
```

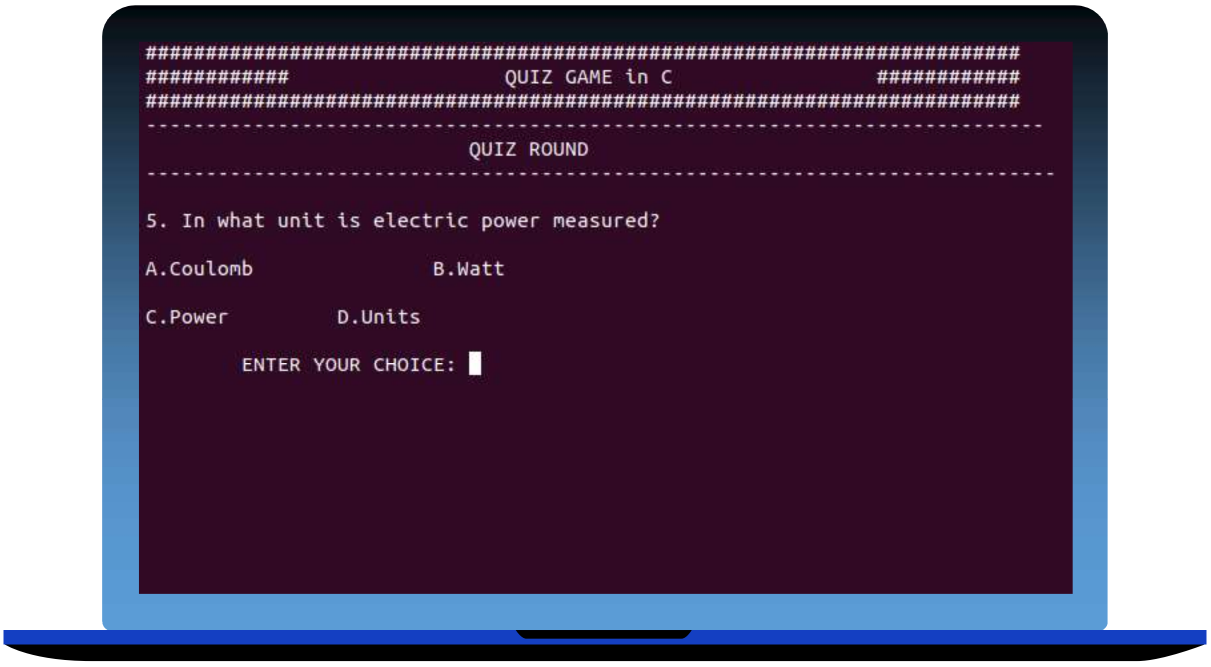
(or)



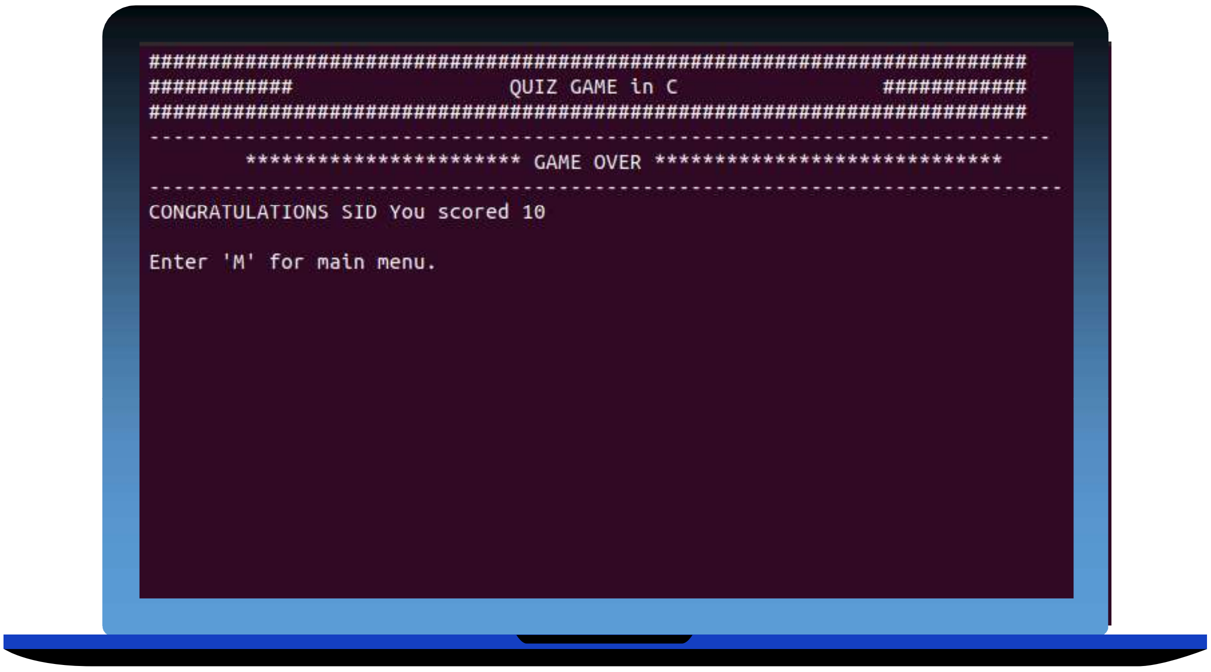
```
##########################################################################
############                 QUIZ GAME in C                 ############
##########################################################################
----------------------------------------------------------------------------
        CONGRATULATIONS. You are eligible to play the Game.
----------------------------------------------------------------------------


!Press any key to Start the Game!:
```



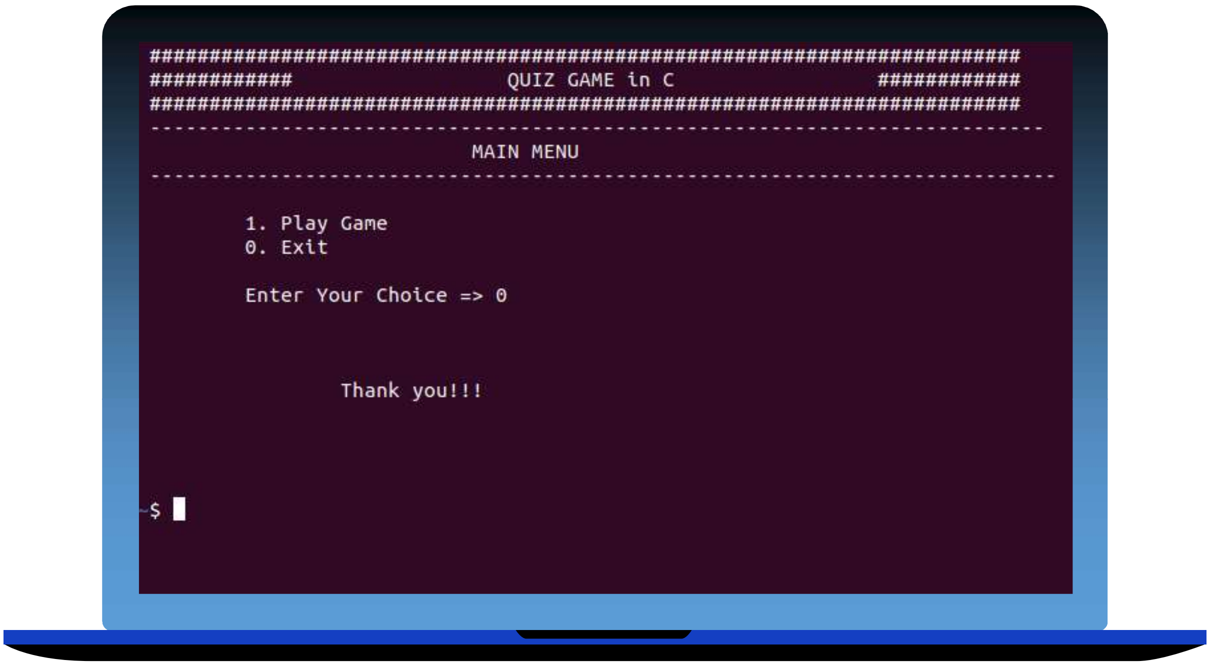
```
##########################################################################
############                  QUIZ GAME in C                  ############
##########################################################################
----------------------------------------------------------------------------
                           QUIZ ROUND
----------------------------------------------------------------------------

1. What is the National Game of England?

A.Football                B.Basketball

C.Cricket                 D.Baseball

        ENTER YOUR CHOICE:
```



```
##########################################################################
############                  QUIZ GAME in C                  ############
##########################################################################
----------------------------------------------------------------------------
                           QUIZ ROUND
----------------------------------------------------------------------------

5. In what unit is electric power measured?

A.Coulomb                 B.Watt

C.Power         D.Units

        ENTER YOUR CHOICE:
```



```
##########################################################################
############                  QUIZ GAME in C                 ############
##########################################################################
---------------------------------------------------------------------------
        *********************** GAME OVER *****************************
---------------------------------------------------------------------------
CONGRATULATIONS SID You scored 10

Enter 'M' for main menu.
```



```
##########################################################################
############                  QUIZ GAME in C                 ############
##########################################################################
---------------------------------------------------------------------------
                           MAIN MENU
---------------------------------------------------------------------------

        1. Play Game
        0. Exit

        Enter Your Choice => 0



                Thank you!!!



~$ 
```

CHAPTER 9

CONCLUSION

Quiz Game Project is a simple console application designed to demonstrate the practical use of C programming language and its features.

This system can be used in Schools/Colleges or vice versa for conducting Quiz contest.

CHAPTER 10

REFERENCES

E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

K R Venugopal | Sudeep R Prasad, Mastering C, Tata McGraw-Hill

https://aticleworld.com/library-management-system-project-in-c/

https://www.geeksforgeeks.org/enumeration-enum-c/