



THAPATHALI CAMPUS
Institute of Engineering

A PROJECT REPORT

ON

“WHO WANTS TO BE A MILLIONAIRE?” GAME

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE
COURSE OF CT 401 COMPUTER PROGRAMMING

BACHELOR OF ELECTRONICS, COMMUNICATION AND INFORMATION
ENGINEERING

SUBMITTED BY

RAJAT DULAL

THA076BEI020

BISHWAMBHAR DAHAL

THA076BEI009

SURAMYA POKHAREL

THA076BEI039

SUBMITTED TO

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

March 2020

DECLARATION

We hereby declare that the project work report entitled **“WHO WANTS TO BE A MILLIONAIRE?”** submitted for the partial fulfillment of the requirements for the course of CT 401 Computer Programming is our original work and the Project Work Report has not formed the basis for the award of any degree, diploma, or other similar titles.

Signature

Name of Student: Rajat Dulal

Signature

Name of Student: Bishwambhar Dahal

Signature

Name of Student: Suramya Pokhrel

ACKNOWLEDGEMENTS

Firstly, we would like to express our deep gratitude to Saroj Shakya sir for letting us to do this wonderful project and providing guidelines. We would like to say thank you to the electronics and computer department of Thapathali Campus for providing us a place to learn and a space to create. We would also like to extend our thanks to the lab technicians and other teachers who directly or indirectly helped us in this project.

We would like to give special thanks to our classmates and pals who encouraged, discussed and exchanged ideas of each other. Without their help, this project could not be imagined. Special gratitude to our seniors for providing us reference materials.

Finally, we would like to thank our parents for encouraging us throughout our study.

ABSTRACT

"Who Wants To Be A Millionaire?" is an internationally popular game-show franchise originally created in England. This game-show is based on a simple format i.e. the more wisely you answer, the more money you'll win. So, based on this show's immense popularity we have decided to make a game of the same format with the same name as a part of our minor project for the first semester.

This proposal contains a detailed description about the game and its various features. The game consists of fifteen questions leading the player from a hundred dollars to one million dollars with the difficulty level increasing with each correct answer. To facilitate the players we have included three lifelines-- Fifty-fifty, Flip the question and Double dip. The user's journey through the questions consists of two marks one at fifth question for a thousand dollars and another at tenth question for thirty-two thousand dollars. The player can quit the game at any point winning the amount of money he currently has in his prize bank. If he gives the wrong answer for any question his prize money is downgraded to the prize money of the mark he has already reached. If he gives the wrong answer before reaching the first mark he doesn't get any prize money and his journey ends. The lifelines, the player has can be used at any question for his aid.

Key Words:

File handling in C; Use of structures in C; Use of Pointer Array in C; PlaySound using different C library; Currency; Validation of User Information.

TABLE OF CONTENTS

DECLARATION	2
ACKNOWLEDGEMENTS	3
ABSTRACT	4
LIST OF FIGURES	7
LIST OF TABLES	8
CHAPTER 1: INTRODUCTION	9
1.1 Error! Bookmark not defined.	9
1.2 Error! Bookmark not defined.	9
1.3 Error! Bookmark not defined.	0
1.4 Error! Bookmark not defined.	0
1.5 Error! Bookmark not defined.	0
1.6 Error! Bookmark not defined.	0-11
1.6.1 Error! Bookmark not defined.	0
1.6.2 Error! Bookmark not defined.	0
1.6.3 Error! Bookmark not defined.	11
1.7 Error! Bookmark not defined.	11
1.7.1 Error! Bookmark not defined.	11
1.7.2 Error! Bookmark not defined.	11
CHAPTER 2: LITERATURE REVIEW	12
CHAPTER 3: DESIGN & METHODOLOGY	13-16
CHAPTER 4: IMPLEMENTATION AND RESULT	17-21
4.1 Error! Bookmark not defined.	17
4.2 Error! Bookmark not defined.	18-21
CHAPTER 5: CONCLUSION & FURTHER WORK	22
5.1 Error! Bookmark not defined.	22

5.2 Error! Bookmark not defined.22

REFERENCES 23

APPENDICES 24

LIST OF FIGURES

FIGURE

PAGE

Figure A- G: ----- 18-21

LIST OF TABLES

TABLE	PAGE
-------	------

Table 1: Gantt Chart-----	22
---------------------------	----

CHAPTER 1: INTRODUCTION

1.1 Background

Quiz has always been popular among every age group of people. General knowledge has been the matter of interest or even a compulsion as it is a must in beauty pageant, competitive exams, and quiz games or even to be generally aware about the different happenings around the world. Considering the demand of students who are preparing for competitive exams like Public service commission(PSC), Nepal Army, etc. where general knowledge is a must, we have thought of this “Who Wants To Be A Millionaire?” program. Apart from that any general knowledge enthusiast would obviously find this game interesting. People are curious to test their general knowledge and understandings through different means. A Quiz is a form of game or mind sport, in which the players attempt to answer general knowledge type questions correctly. It is a game to test your knowledge in different subjects. Generally quiz is played among the teams but single player quiz game program has been popular round the globe. The evaluation is made on the basis of number of right answers the player answers. The quest of making this quiz game more interactive and user friendly, has led us to the introduction of many life lines in the game to assist the player in answering the right answer. The failure of answering the correct answer leads to the over of the player’s journey in the game.

Various quiz games which made using C Language are available in Internet but our Quiz game is the most user-friendly, interesting and logical one you can find. Our project stands out as one of the most interactive platform of Quiz Game history. To give user an option to enjoy our game for several times and compare his/her progress each time, we have provided the option of saving the player’s past progress/score in a text file using File Handling technique in C programming Language. Similarly, For the purpose of providing a good experience in playing the game we have added some audios to our game. Hence, Our project expects to meet the full utilization of the resources as provided by the C language.

1.2 Problem Statement

To provide user the best quiz game experience, introducing 3 lifelines, different sound effects and different difficulty level for each question in a single game was literally a big challenge for us. We brought the concept of quiz game to a completely new level. Not just that but manipulating pointers and arrays to select the random questions from array and saving the player’s score after each game in a text file was also a great challenge for us. The design of mechanism for validation of a player’s personal information is also quite complex.

1.3 Objectives

1. To develop a prototype of a popular game-show to provide ordinary people simulation of playing it.
2. To learn algorithms for displaying random questions in a controlled way.

1.4 Applications

1. It can be highly applicable for those students who are planning to attend competitive exams (Loksewa and Other) where general knowledge is a must.
2. It can be used by students who want to increase their general knowledge in an interactive way.

1.5 Project Features

1. It provides an interactive platform for general knowledge enthusiast.
2. It provides career mode that is developed using file handling that provides a completely innovative way to play quiz.
3. It has made a complete use of the resources provided by the C language and made an integrated project.
4. It has a user friendly interface.

1.6 Feasibility Analysis

1.6.1 Economic Feasibility

This program is economically feasible. It is an open source as well as a simple program it doesn't cost much for the program.

1.6.2 Technical Feasibility

It works on the feature of C like library functions and file handling. The required activities are handled by the compiler of Code Blocks, Turbo C or other programs.

1.6.3 Operational Feasibility

All we need is a computer with code blocks or turbo C with GNU GCC compiler along with –lwinmm linker.

1.7 System Requirement

It doesn't need any additional hardware or software to operate the program but the following requirement should be strongly maintained.

1.7.1 Software Requirement

System requirement: windows 98 or higher

Code blocks along with its all compiler files need to be installed and a linker –lwinmm should strictly be added.

1.7.2 Hardware Requirement

It doesn't need any additional hardware or software to operate the program but the following requirement should be strongly maintained.

800 MHz processor or above

20 MB of hard disk space

512 MB of RAM or higher

CHAPTER 2: LITERATURE REVIEW

2.1. Historical Background

"Who Wants To Be A Millionaire?" first debuted in England on 4 September 1998 on the ITV network and was aired until its final episode on 11 February 2014. The format of this show was created by David Briggs, Mike Whitehill and Steven Knight. It was produced by David Briggs and he chose Chris Tarrant as its host. It was designed in such a way so as to create some tension in the player and viewers with the tones, music and animations in order to make it more thrilling.

2.2. Legacy of "Who Wants To Be A Millionaire"

This show was well received among audiences and inspired many international variants of this show like "Kaun Banega Crorepati" in India, "Wer wird Millionär" in Germany, "Ko Banchha Crorepati" in Nepal and so on. ITV revived the series at its twentieth anniversary airing seven episodes from 5 to 11 May 2018 which was very well received by critics and fans. Many games and miniclips were also inspired by this show including the one we are proposing to make.

2.3. Significance

"Who Wants To Be A Millionaire?" hugely affected the modern culture especially of '90s and early 2000s' kids. The show since its debut has always been a household name. Everyone has at least once dreamt of going to this show and winning the top prize. This show made people realize, at least up to some degree, that being intelligent was rewarding and useful. It also changed lives of many ordinary people who went there as contestants. Many movies including the Oscar winning "Slumdog Millionaire" was inspired by this show. Hence, to provide people some sense of gratification and test their intelligence on random fields we have decided to make this game and present it as our minor project.

CHAPTER 3: DESIGN & METHODOLOGY

This program usually consists of three main parts (excluding some little consoles):- The sign in, The gameplay and The Scoreboard.

3.1 Information validation/Sign in

At the start of the game user is asked to enter his information i.e. his name, age and e-mail address which is stored in a structure called *personal_detail*. Now the validity of the provided information is checked using a function named *person_info()*.

3.1.1 Validation of name: -

At the beginning, the user is asked to enter his name which must include his first and last names and may include his middle name. To check the validity of the name that the user has entered, we first find out the length of the name by using *strlen()* function. Then with the help of a loop, we access each character of the name and count the number of white spaces using *If* statement. If the number of white spaces exceeds two or remains zero then the user is asked to enter his name again. This process continues until the user enters a valid name.

3.1.2 Validation of age:-

The user is also asked to enter his age at the start of the game. His age is considered valid if it ranges between 0 and 110. The age is checked by using *If* statement. If the age doesn't fall in that range, the user is asked to re-enter his age until he enters a valid age.

3.1.3 Validation of email:-

At the beginning, the user is also asked to enter his e-mail address. The e-mail he enters is considered valid, if it contains both '@' and '.' without their repetition. To check its validity we first find out the length of the email by using *strlen()* function. Then with the help of a loop, we access each character of the e-mail and count the number of '@'s and '.'s using *If* statement. If their number is not equal to 1 each then the user is asked to enter his e-mail address again. This process continues until the user enters a valid e-mail.

3.2 The Game play

3.2.1 Storing the Data

All the questions, answers and descriptions are stored in respective string arrays and right answers are stored in a character array. All these information are passed to a function named `solution_calulation` when called.

3.2.2 Body

This is the main section of the program. The body section is defined in the function: `solution_calculation()`. This section plays a major role on controlling the flow of the program. The function takes all the values of questions, options, right answers and descriptions defined on the `ques()` function and then does the following tasks :-

I – Firstly, it generates a random number by using `rand()` function and its modulo 5 value is stored on a variable which is further used as the index for other arrays except the *ans* array. For *ans* array's index we use certain mathematical calculations to only access a particular option from the 'ans' array.

II – Secondly, we display a random question of the particular category with its respective options and then ask the user for its answer. The answer is stored in uppercase with use of `toupper()` function.

III- Checking of answer: In this section of body we check the answer given by the player for its correctness. For this *if* statement is used. If the player's answer and the right answer (stored in an array) match, the user gains points and proceeds to the next question and if it doesn't match then `score()` function is called, which brings the game to its end.

The user can also use life-lines during the gameplay for his aid by entering characters: X, Y and Z for double-dip, fifty-fifty and flip question respectively. If user inputs any of the above mentioned characters then firstly the program checks if it's the first time that life-line is being used. For this, we use a static variable whose value changes from zero after the particular life-line is used. If that particular life line is being used for first time, then the function of life-line is called else its use is restricted and the same question is displayed again.

3.3.3 Life lines

Three life-lines are provided for the player's aid. All the life-lines are kept in different functions named as *lifeline1()*, *lifeline2()* and *lifeline3()*.

I- Double Dip(2X):-

When player enters the character 'X' instead of the answer then double-dip life-line is activated. Double-dip lifeline is defined under the function *lifeline1()*. This function takes the variables of question, answer, right answer, description and random number from the function *solution_calculation*. Now the user is asked for his answer. If the answer is right then the user proceeds to the next question and if answer is wrong, the user is asked to input another answer. If the second answer is right then the user again proceeds to the next question but if answer is wrong again then the game comes to an end. For this *If – else if* statement is used.

II- Fifty Fifty

If player enters the character 'Y' instead of the answer then fifty-fifty life-line is activated. Fifty-fifty life line is defined under function *lifeline2()*. This function takes the variable of question, answer, right answer, description and random number from the function *solution_calculation*. At first, right answer is stored in a variable *corr* and then a *switch-case* statement is used to assign specific number to a variable *p* to know the correct option for latter processes. Then we generate a random number under modulo 5 in a looping statement. Under the looping statement, we use *If* statement which checks if the option of the index of random number is not empty or variable *p* is not equal to the newly generated random number. If the condition is true then it changes the option having index of the random number to empty. The loop is continued until two options are removed. Then question is displayed with two options removed and user is asked to enter his answer. The answer is checked using *If* statement.

III- Flip the question:

If player enters the character 'Z' instead of the answer then flip the question life-line is activated. Flip the question life-line is defined under function named *lifeline2()*. This function takes the variable of question, answer, right answer, description and random number from the function *solution_calculation*. Now new random number of modulo 5 is generated under a loop. Then *If* statement is used to check if the newly generated random number is equal to the random number generated in *solution_calculation* or not. If it is equal, then again a new random number is generated under a loop and the same process is repeated until newly generated random number is not equal to the random number generated in *solution_calculation*. Now newly generated random number is used as the index to select a new question of same category.

3.3 The Scoreboard

File Handling in C technique is used to save the personal information (Player's Name, Email, Age) and score of each and every game played in a text file named "ScoreList". This stored data can also be displayed at the end of each game by using read "r" option or it can be easily accessed by opening the text file from its destination location.

CHAPTER 4: IMPLEMENTATION AND RESULT

4.1. Implementation detail

The functions used in the program as to maintain the program simpler and make a better coordination among the team members are as follows:

1. `main()`:-
2. `lifeline1()`:- This function is used for 2X lifeline.
3. `lifeline2()`:- This function is used for 50-50 lifeline.
4. `Lifeline3()`:- This function is used for Double Dip lifeline.
5. `solution_calculation()`:- This function is used as a reference function to check for correct answer, provide option for accessing lifelines,etc.
6. `saving()`:- This function is used for saving the player's personal information and player's score in a textfile.
7. `score()`:- This function is used to display final report card.
8. `person_info()`:- This function is used to ask input of player's personal information and check its validity.

Note:-

1. Fifteen `ques()` function used in the program from `ques1()` to `ques15()` are used to store questions, answers and their descriptions.
2. A structure `personal_detail` is also used to support `person_info()` function.

Header files included in our programs are as follows:

#include <stdio.h>: It is basically a header file, which **means** that while writing the program, we are trying to import some predefined library function in order to enhance our program. **stdio:** means Standard Input/Output.

#include <conio.h>: It is C header file mostly used by compilers to provide console input/output.

#include <windows.h>: This header file, is a base header file for WIN32 programming, its contain deceleration for almost all basic windows macros and different typedef.

#include <string.h>: the header in the C standard library for the C programming language which contains macro definitions, constants and declarations of functions and types used not only for **string** handling but also various memory handling function

#include <time.h>: It is a header file defined in the C Standard Library that contains time and date function declarations to provide standardized access to time/date manipulation and formatting.

#include<stdlib.h>: Stdlib.h contains the standard library definitions. delay, calloc, malloc, realloc, system, etc. are the functions used under the library file.

#include<dos.h>: -Dos.h header file of C language contains functions for handling interrupts, producing sound, date and time functions etc.

#include<ctype.h>: The **ctype.h** header file of the C Standard Library declares several functions that are useful for testing and mapping characters.

#include<MMSystem.h>:-MMSystem.h header file of the C standard Library contains functions for reading a audio sound in .wav format.

4.2 Result anlaysis

The purpose of our C program is to provide user an interface to play interactive and interesting quiz game. The first window is the console window which displays the creator's names, rules on how the game should be played and accepts personal information from the user and makes access to Game Play after few sound effects.

```

WELCOME TO THE GAME 'WHO WANTS TO BE A MILLIONAIRE?'

Created by:
Rajat Dulal
Bishwambhar Dahal
Suramya Pokharel

>>>>>>>>>GAME INSTRUCTIONS<<<<<<<<<<

1. 'Who wants to be a Millionaire?' is a quiz game in which a player has to answer 15 questions correctly(one after another) to be a millionaire.
2. After Entering the answer the player cannot change his/her answer.
3. To each question the player gets 4 options among which he/she should choose the correct answer.
4. To answer the question, the players is only allowed to type either 'A' for first option, 'B' for second option, 'C' for third option and 'D' for fourth option.
5. For the aid of player in answering the questions, there are three lifelines available in the game- 50-50, Flip the question and Double Dip.
6. Player can use each lifeline only one time in a game.
7. The alphabets are Case-Insensitive.
8. Player is not allowed to use 2 lifelines in the same question.

WARNING: DONOT ENTER ANY OTHER ALPHABETS WHILE ANSWERING THE GAME UNLESS YOU WANT YOUR GAME TO QUIT.

Enter your Full name      :Rajat Dulal

Enter your Age    :20

Enter your Email Address      :ritdulal@gmail.com

```

Figure A.

After the Validation of the personal information of the player, The game starts by displaying its first question after few sound effects and pressing any key.

```
Your question is:
1.Complete this common saying: A stitch in time,_____.

      A: Reduces pain      B: Saves nine
      C: Heals wounds      D: Prevents risk of infection

For life line
*Type X for 2X
*Type Y for 50-50
*Type Z for flip the question

Enter your answer: _
```

Figure B.

If the player enters the correct answer, the player earns some currency, the difficulty level of the question rises and the player reaches to the next level question. While answering the question, the player can even take the help of lifelines that we have provided.

Display of 2X Lifeline:

```
/* Your 2X life line has been activated.

Your question is:
1.Complete this common saying: A stitch in time,_____.
      A: Reduces pain      B: Saves nine
      C: Heals wounds      D: Prevents risk of infection

Enter your answer: A
Wrong Answer.Please Try Again.

Enter your next answer: B_
```

Figure C.

As displayed above the player is allowed to enter 2 answers even after the player makes the first answer mistake.

Display of 50-50 lifeline:

```
/* Your 50 - 50 life line has been activated.  
  
Your question is:  
2.What is the population of the Earth as of 2017?  
    A:      B: 7.53 billion  
    C:      D: 8.63 billion  
  
Enter your answer: _
```

Figure D.

As you can see, the two wrong answers out of four has been removed after using 50-50 lifeline.

Display of Double Dip lifeline:

```
/* Your flip question life line has been activated.  
  
Your question is:  
1.Which of the following countries lie in Asia?  
  
    A: Mexico      B: Libya  
    C: Serbia      D: Nepal  
  
Enter your Answer:
```

Figure E.

As you can see, the question number 1 has been flipped to another question.

After the player answers the question the description of each question and answer is shown as follows:

```
Enter your Answer: d
Correct answer

Mexico lies in North America, Libya in Africa and Serbia in France.
Nepal lies in Asia and is popular for the highest peak in the world
i.e Mt. Everest. It lies between India and China.
```

Figure F.

This process continues till the player answers all the questions correctly upto 15 questions. After the correct answering of 15th question or after answering wrong in any question, the following console/Scoreboard is displayed and the game quits after pressing any key accompanied by some sound effects.

```
>>>>>>>>YOUR PROGRESS<<<<<<<<<<

Player's Name: Gs
Player's Age: 2
Player's Email: @.
You Won: $100
Total Time taken is 8 min and 13 sec.

>>>>>>>>THANK YOU FOR PLAYING THIS GAME.<<<<<<<<<<

Press any key to exit.
```

Figure G.

CHAPTER 5: CONCLUSION & FURTHER WORK

5.1 Conclusion

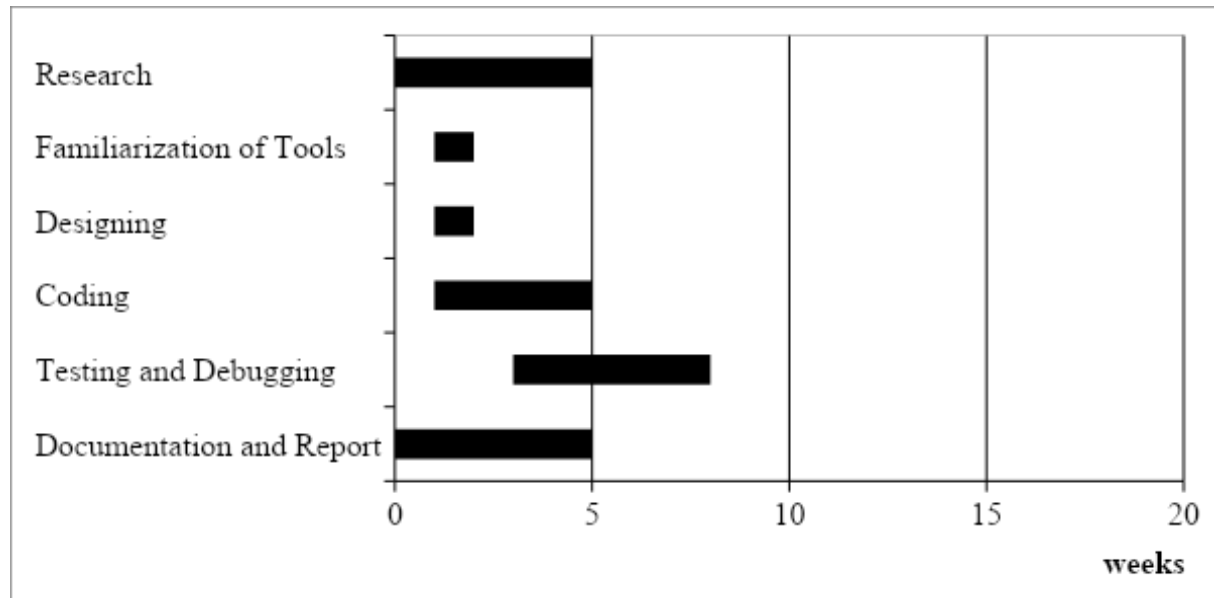


Table 1: Gantt chart

5.2 Further works

We made this project with all our ideas, knowledge and wisdom but It still lacks certain where we can work on. We tried to toggle the option as we had done to the questions so as to make it more interesting which hasn't been completed for now. Similarly, We have not used graphics in this program due to several constraints. Without using graphics, too, we have tried to provide good interface to users. Still, there are many things to be improved.

REFERENCES

<Should be in proper format(follow the IEEE format) and each of the references must be cited somewhere in proposal body.... citation must be done in order... ie reference [1] cannot be after [2] in the proposal> example reference format is given below... Note that you may also include Bibliography Section which is similar to reference but is not cited in the main text... <visit <http://www.citethisforme.com/ieee> >

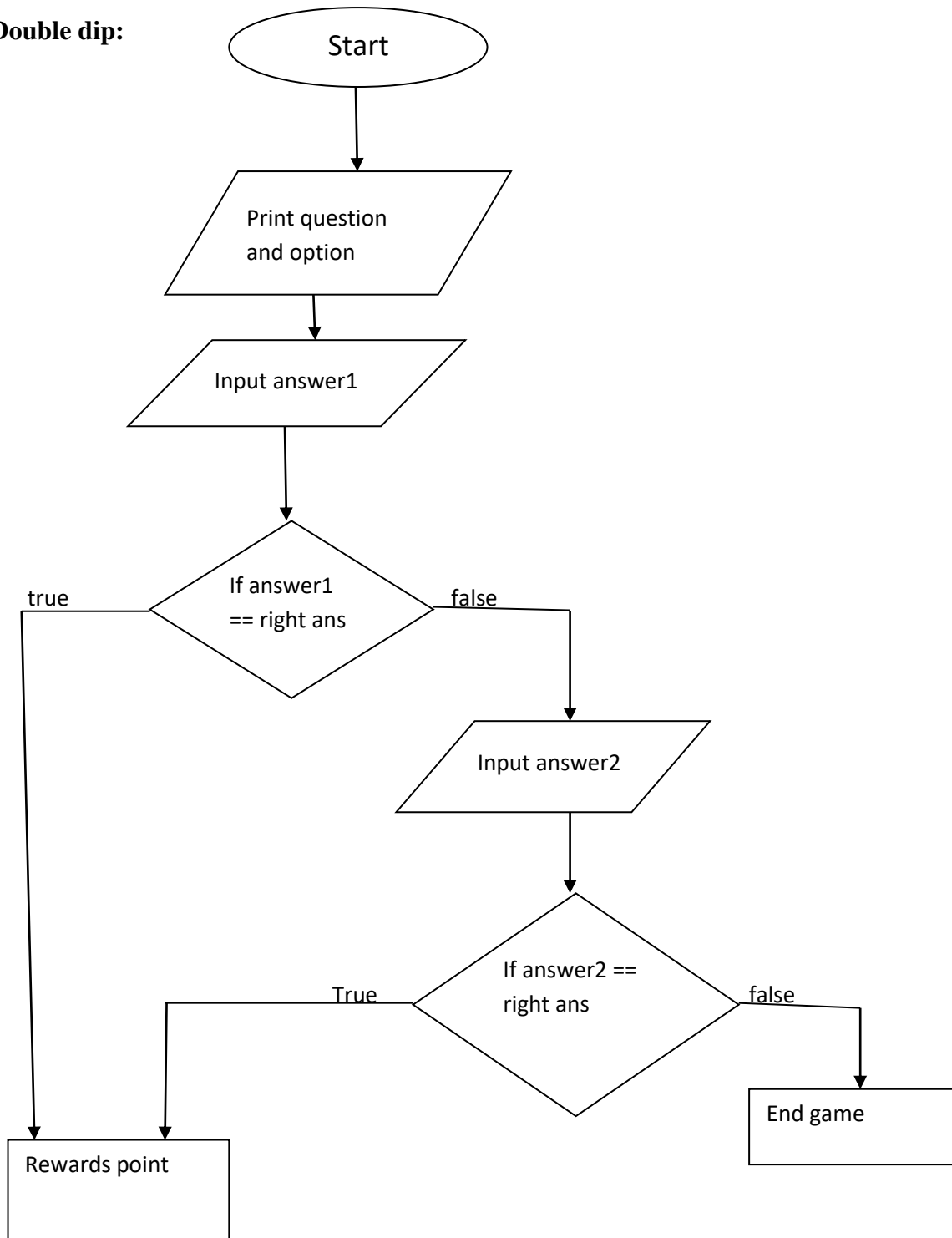
- [1] M. Naor and A. Shamir, "Visual cryptography", *Eurocrypt 1994, Lecture Notes in Computer Science*, vol.950, pp. 1-12, Springer-Verlag, 1994
- [2] Z. Zhou, G. R. Arce and G. Di Crescenzo, "Halftone visual cryptography", *IEEE Trans. Image Process.*, vol. 15, no. 8, pp.2441-2453, 2006
- [3] Z. M. Wang, G. R. Arce, and G. Di Crescenzo, "Halftone visual cryptography via direct binary search", *Proc. EUSIPCO'06, Florence, Italy, Sep. 2006*
- [4] T. M. Alkharobi, A. K. Alvi, "New Algorithm For Halftone Image Visual Cryptography", *IEEE 2004*
- [5] Y.-C. Hou, "Visual cryptography for color images", *Pattern Recognition*, vol. 36, iss. 7, pp. 1619–1629, 2003
- [6] M. Nakajima and Y. Yamaguchi, "Extended visual cryptography for natural images," *Proc. WSCG Conf. 2002*, pp. 303–412, 2002
- [7] Y.-C. Zeng and C.-H. Tsai "Controllable transparency image sharing scheme for grayscale and color images with unexpanded size", *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacif*, pp. 1-4, 2013

APPENDICES

The Step-Wise Flowchart for our program “Who Wants To Be A Millionaire?” can be summarized as shown below:

Flowcharts for Lifelines:

1) Double dip:



2) Flip question:

