

Project On  
**Hangman Game**

**GROUP NUMBER:** 03

**GROUP MEMBERS:**

* Rafa Tasnim : 2019-1-60-081
* Malyha Bintha Mabud : 2019-1-60-128
* Humaira Anan Neela : 2019-1-60-218

**DEPARTMENT:** Computer Science and Engineering

**SESSION:** Summer 2021

**SECTION:** 02

**SUBMITTED TO:**

Dr. Ahmed Wasif Reza

Associate Professor

East West University

**SUBMITTED DATE:** 16/09/2021

**PROJECT EVALUATION RUBRICS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Max.** |  | **Awarded** |
|  | |  |  |  |
| **A. Report** | |  |  |  |
|  |  |  |  |  |
| i. | Introduction / Problem statement |  |  |  |
|  |  |  |  |  |
| iv. | System Design |  |  |  |
|  |  |  |  |  |
| v. | Program output (Screen shots) |  |  |  |
|  |  |  |  |  |
| vi. | Source code |  |  |  |
|  | |  |  |  |
| vii. Disk/CD neatly attached (Y/N) | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **B. Source Code** | |  |  |  |
|  |  |  |  |  |
| i. | Style |  |  |  |
|  |  |  |  |  |
|  | Indentation |  |  |  |
|  |  |  |  |  |
|  | Self-documentation |  |  |  |
|  |  |  |  |  |
| ii. | Modularity (small size functions) |  |  |  |
|  |  |  |  |  |
| iii. | Error reporting capabilities |  |  |  |
|  |  |  |  |  |
| iv. | Code efficiency, strategy, and originality |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **C. Program Execution** | |  |  |  |
|  |  |  |  |  |
| i. | Compile without errors |  |  |  |
|  |  |  |  |  |
| ii. | User friendly |  |  |  |
|  |  |  |  |  |
| iii. | Error free during runtime |  |  |  |
|  |  |  |  |  |
| iv. | Program output |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **D. Presentation and Demonstration [Psychomotor Domain]** | |  |  |  |
|  | |  |  |  |
| i. Presentation and communication skills (**Soft skill)** | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **E. Bonus** | |  |  |  |
|  |  |  |  |  |
| i. | Extra significant features |  |  |  |
|  |  |  |  |  |
|  | **TOTAL** |  | **11** |  |
|  |  |  |  |  |

**PROJECT DECLARATION:**

**(Student 1)**

**Student ID 2019-1-60-081**

**Name Rafa Tasnim**

**Session Spring-2019**

**Project No. 03**

**Date submitted 16-9-2021**

**Deadline of the project 16-9-2021**

**My contribution in doing this 33.33%**

**project (in percentage) in the group**

**Description of my contribution in this get\_word()**

**project in the group get\_input()**

**Number of hours I spent in doing this Few days**

**project**

**(Student 2)**

**Student ID 2019-1-60-128**

**Name Malyha Bintha Mabud**

**Session Spring-2019**

**Project No. 03**

**Date submitted 16-09-2021**

**Deadline of the project 16-09-2021**

**My contribution in doing this 33.33%**

**project (in percentage) in the group**

**Description of my contribution in this draw\_platform()**

**project in the group print\_blanks()**

**Number of hours I spent in doing this Few days**

**project**

**(Student 3)**

**Student ID 2019-1-60-218**

**Name Humaira Anan Neela**

**Session Spring-2019**

**Project No. 03**

**Date submitted 16-09-2021**

**Deadline of the project 16-09-2021**

**My contribution in doing this 33.33%**

**project (in percentage) in the group**

**Description of my contribution in this start \_game()**

**project in the group get\_word()**

**Number of hours I spent in doing this Few days**

**project**

We hereby certify that this project represents the work done by all our group members with our contribution clearly stated above without copying from any other resources. We declare that no part of our work has been copied from or by other groups, and that no collusion has taken place with any other persons or groups.

We certify that any disks submitted with this project have been virus checked and have no viruses on them.

**INTRODUCTION/PROBLEM DEFINITION:**

The project we have been working on is one of the most popular game called “HANGMAN”. The systems of the game are very simple. Here we would have to guess a word which will be randomly selected. On the process we have to guess letter by letter. If we are able to choose the correct letter then we will be able to choose the next one and if we failed in choosing the correct one then the process of hanging the poor man will continue step by step. We will be given certain number of chances and we have to guess the letter correctly if we failed the time of chances then the man will be hanged. In the game 10 score will be added for every correct word guess and we can continuously play as many rounds as we want. Here we are going to use FILE as saving some sample words from where the word will be randomly chosen. Here we have 10 attempts for the easy level,6 chances for medium level and 4 hard attempts for hard level. After few wrong assumptions console will also show some hints for guessing the write word.

**SYSTEM DESIGN/FLOW CHART:**

**Display Messages**

* **Easy Level**
* **Medium Level**
* **Hard Level**

**While (1)**

**Level**

**End Game**

**Lose Message**

**Draw Platform**

**Get Word**

**Start Game**

**Attempt > 0**

**Draw Platform**

**Print Blank**

**Print Blank**

**Get Input**

**Draw Platform**

**Win Message**

**Attempt = 0**

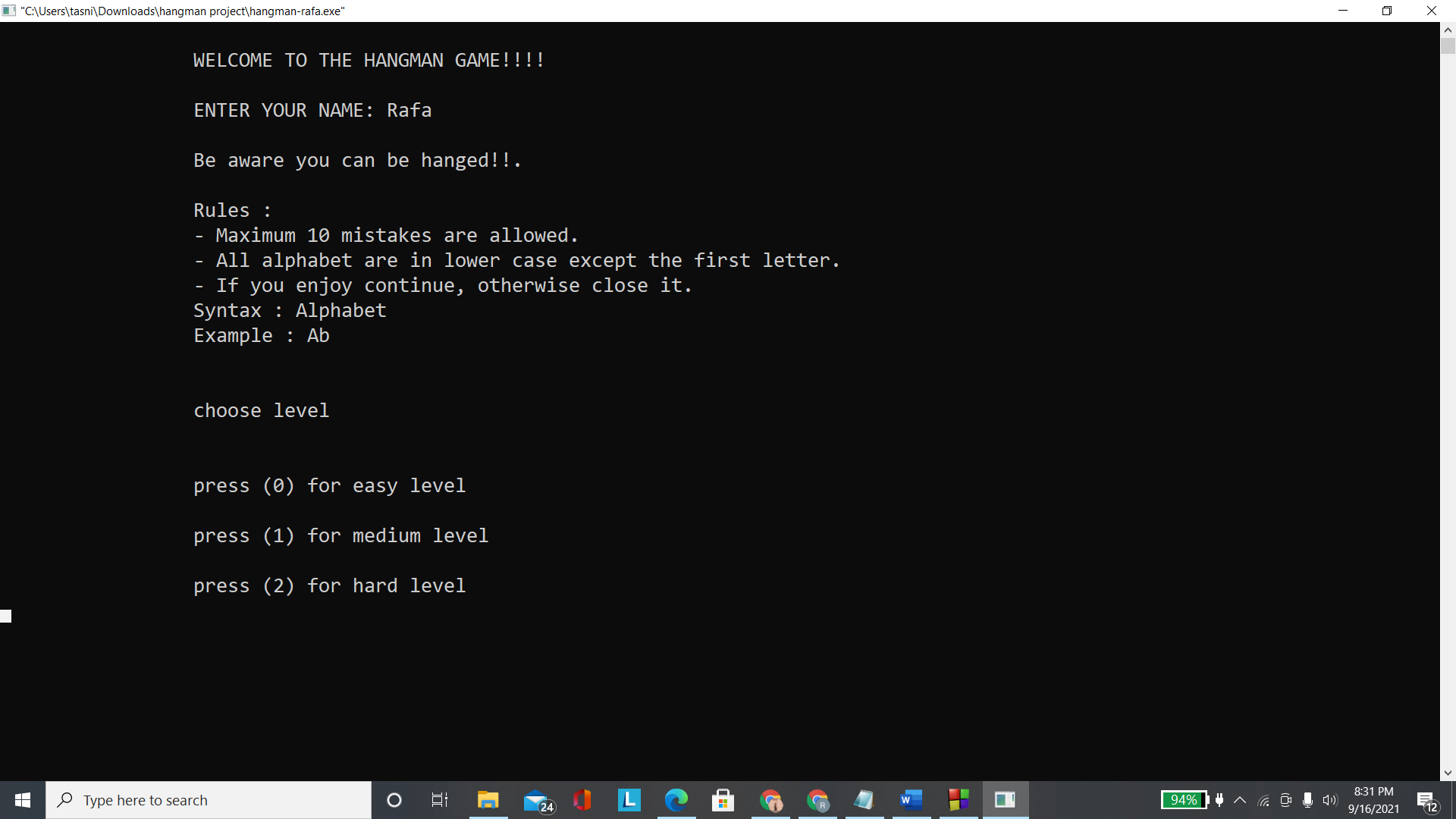
**Incorrect Guess(attempts- -)**

**Correct Guess**

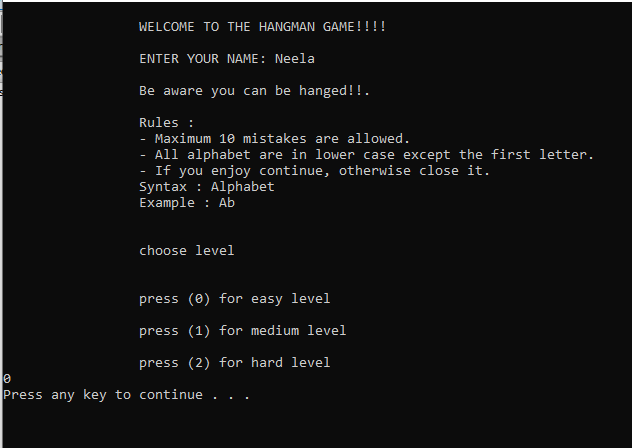


**If user continues**

**PROGRAM OUTPUT/SCREEN SHOT:**

**Display Screen:** 

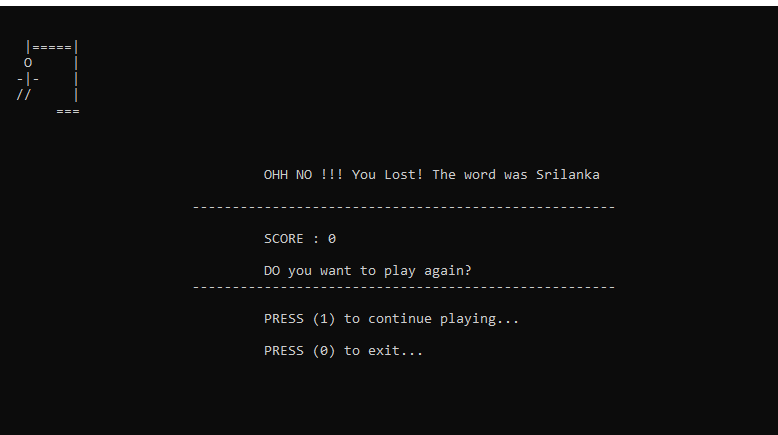
**Level 0: Easy (Easy Level)**

****

**Easy Level Win Messages:**

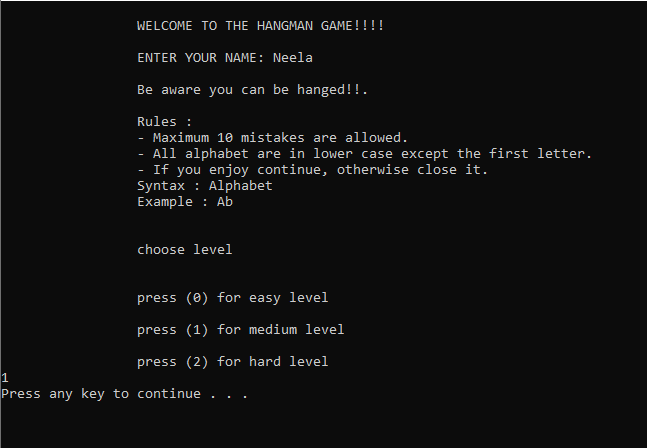
****

**Easy Level Lost Messages:**

****

**Level 1: (Medium Level)**

**Medium Level Choice:**



**Medium Level Win Messages:**

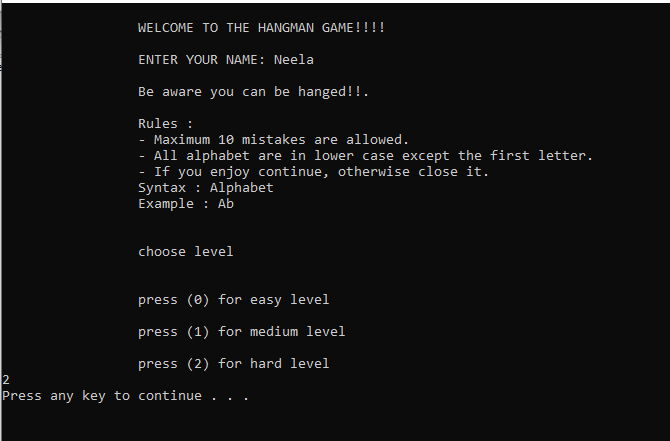


**Medium Level Lost Messages:**



**Level 2: (Hard Level)**

**Hard Level Choice:**



**Hard Level Win Messages:**



**Hard Level Lost Messages:**



Saved Files:



**LIMITATIONS OF YOUR PROGRAM:**

Sometimes when we run the program it will take some time to get start and show the console screen.

**APPENDIX (SOURCE CODE):**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <string.h>

char word[100];

char guessed\_letters[100];

char user\_guess[] = "";

char blank[1] = "-";

int random\_number;

int word\_len;

int user\_input;

int attempts;

int score = 0;

int total\_score;

int level;

int chance =6;

int hard\_attempt=4;

int attempt\_count=10;

int chance\_count=6;

char name[50];

void start\_game();

void get\_input();

void print\_blanks();

void draw\_platform();

void draw\_medium\_platform();

void draw\_hard\_platform();

void get\_word();

void get\_wordM();

void get\_wordH();

void winMsg();

void looseMsg();

int main()

{

system("cls");

printf("\n\t\t WELCOME TO THE HANGMAN GAME!!!!\t\t\n");

printf("\n\t\t ENTER YOUR NAME: ");

scanf("%s", name);

printf("\n\t\t Be aware you can be hanged!!.");

printf("\n\n\t\t Rules : ");

printf("\n\t\t - Maximum 10 mistakes are allowed.");

printf("\n\t\t - All alphabet are in lower case except the first letter.");

printf("\n\t\t - If you enjoy continue, otherwise close it.");

printf("\n\t\t Syntax : Alphabet");

printf("\n\t\t Example : Ab \n\n");

printf("\n\t\t choose level \n\n ");

printf("\n\t\t press (0) for easy level\n");

printf("\n\t\t press (1) for medium level\n");

printf("\n\t\t press (2) for hard level\n");

scanf("%d",&level);

system("PAUSE");

while(1)

{

if(level==0)

{

printf("YOU HAVE CHOOSEN EASY LEVEL \n");

if(level==0)

start\_game();

while (attempts > 0)

{

system("cls");

if (attempts == 5)

printf("\t\t HINT: The word is any Asian country name!!!!\n \t\tkeep trying\n");

if (strlen(guessed\_letters) == word\_len - 1)

{

print\_blanks();

break;

}

else

{

printf("Attempts Remaining: %i\n", attempts);

print\_blanks();

get\_input();

}

attempt\_count--;

}

system("cls");

if (attempts > 0)

{

print\_blanks();

winMsg();

}

else

{

draw\_platform();

looseMsg();

}

}

if(level==1)

{

printf("YOU HAVE CHOOSEN MEDIUM LEVEL \n");

start\_game();

while (chance > 0)

{

system("cls");

if (chance == 3)

printf("\t\t HINT: The word is any European country name!!!!\n \t\tkeep trying\n");

if (strlen(guessed\_letters) == word\_len - 1)

{

print\_blanks();

break;

}

else

{

printf("Attempts Remaining: %i\n", chance);

print\_blanks();

get\_input();

}

chance\_count--;

}

system("cls");

if (chance > 0)

{

print\_blanks();

winMsg();

}

else

{

draw\_medium\_platform();

looseMsg();

}

}

if(level==2)

{

printf("YOU HAVE CHOOSEN HARD LEVEL \n");

start\_game();

while (hard\_attempt > 0)

{

system("cls");

if (strlen(guessed\_letters) == word\_len - 1)

{

print\_blanks();

break;

}

else

{

printf("Attempts Remaining: %i\n", hard\_attempt);

print\_blanks();

get\_input();

}

}

system("cls");

if (hard\_attempt > 0)

{

print\_blanks();

winMsg();

}

else

{

draw\_hard\_platform();

looseMsg();

}

}

scanf("%i", &user\_input);

if (user\_input == 1)

{

continue;

}

if (user\_input == 0)

{

system("cls");

printf("\n\n\t\t\t\t\t THANK YOU!! ");

system("PAUSE");

return 0;

}

}

}

void winMsg()

{

score = 10 + score;

printf("\n\n\t\t\t\t CONGRATULATION!!! %s You Won!\n",name);

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

printf("\n\n\t\t\t\t SCORE : %d", score);

printf("\n\n\t\t\t\t DO you want to play again? \n\n");

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

printf("\n\n\t\t\t\t PRESS (1) to continue playing... ");

printf("\n\n\t\t\t\t PRESS (0) to exit... ");

}

void looseMsg()

{

printf("\n\n\t\t\t\t OHH NO !!! You Lost! The word was %s\n", word);

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

printf("\n\n\t\t\t\t SCORE : %d", score);

printf("\n\n\t\t\t\t DO you want to play again? \n");

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

printf("\n\n\t\t\t\t PRESS (1) to continue playing... ");

printf("\n\n\t\t\t\t PRESS (0) to exit... ");

}

void start\_game()

{

if (level ==0)

{

get\_word();

word\_len = strlen(word);

for (int i = 0; i < sizeof(guessed\_letters); i++)

{

guessed\_letters[i] = 0;

}

attempts = 10;

}

if (level==1)

{

get\_wordM();

word\_len = strlen(word);

for (int i = 0; i < sizeof(guessed\_letters); i++)

{

guessed\_letters[i] = 0;

}

chance = 6;

}

if (level==2)

{

get\_wordH();

word\_len = strlen(word);

for (int i = 0; i < sizeof(guessed\_letters); i++)

{

guessed\_letters[i] = 0;

}

hard\_attempt=4;

}

}

void get\_input()

{

int i;

int letter\_hit = 0;

printf("\nGUESS THE WORD : \n");

scanf(" %c", user\_guess);

for (i = 0; i < word\_len; i++)

{

if (user\_guess[0] == word[i])

{

guessed\_letters[i] = user\_guess[0];

letter\_hit++;

}

}

if (letter\_hit > 0)

{

return;

}

else

{

attempts--;

chance--;

hard\_attempt--;

}

}

void print\_blanks()

{

int i, j;

if(level==0)

draw\_platform();

if(level ==1)

draw\_medium\_platform();

if(level==2)

draw\_hard\_platform();

for (i = 0; i < word\_len; i++)

{

printf("%c", guessed\_letters[i]);

printf(" ");

}

printf("\n");

for (j = 0; j < word\_len - 1; j++)

{

printf("%s", blank);

printf(" ");

}

printf("\n");

}

void draw\_platform()

{

char \*platform[] =

{

" ===\n",

" |\n"

" |\n"

" |\n"

" ===\n",

" =====|\n"

" |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" | |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" |- |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" | |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" // |\n"

" ===\n"

};

if (attempts == 9)

printf("\n\n%s\n", platform[0]);

if (attempts == 8)

printf("\n\n%s\n", platform[1]);

if (attempts == 7)

printf("\n\n%s\n", platform[2]);

if (attempts == 6)

printf("\n\n%s\n", platform[3]);

if (attempts == 5)

printf("\n\n%s\n", platform[4]);

if (attempts == 4)

printf("\n\n%s\n", platform[5]);

if (attempts == 3)

printf("\n\n%s\n", platform[6]);

if (attempts == 2)

printf("\n\n%s\n", platform[7]);

if (attempts == 1)

printf("\n\n%s\n", platform[8]);

if (attempts == 0)

printf("\n\n%s\n", platform[9]);

}

void draw\_medium\_platform()

{

char \*platform\_medium[] =

{

" |=====|\n"

" |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" |- |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" | |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" // |\n"

" ===\n"

};

if (chance == 5)

printf("\n\n%s\n", platform\_medium[0]);

if (chance == 4)

printf("\n\n%s\n", platform\_medium[1]);

if (chance == 3)

printf("\n\n%s\n", platform\_medium[2]);

if (chance == 2)

printf("\n\n%s\n", platform\_medium[3]);

if (chance == 1)

printf("\n\n%s\n", platform\_medium[4]);

if (chance == 0)

printf("\n\n%s\n", platform\_medium[5]);

}

draw\_hard\_platform()

{

char \*platform\_hard[] =

{

" |=====|\n"

" |\n"

" |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" | |\n"

" ===\n",

" |=====|\n"

" O |\n"

" -|- |\n"

" // |\n"

" ===\n"

};

if(hard\_attempt==3)

printf("\n\n%s\n", platform\_hard[0]);

if(hard\_attempt==2)

printf("\n\n%s\n", platform\_hard[1]);

if(hard\_attempt==1)

printf("\n\n%s\n", platform\_hard[2]);

if(hard\_attempt==0)

printf("\n\n%s\n", platform\_hard[3]);

}

void get\_word()

{

FILE \*fp,\*fp1,\*fp2;

int line\_number = 0;

int i;

char current\_word[100];

fp = fopen("easy.txt", "r");

if (fp == NULL)

{

printf("Error in opening file");

exit(0);

}

while (fgets(current\_word, 50, fp) != NULL)

{

line\_number++;

}

random\_number = rand() % line\_number;

rewind(fp);

for (line\_number = 0; line\_number != random\_number; line\_number++)

{

fgets(current\_word, 50, fp);

}

strcpy(word, current\_word);

fclose(fp);

}

void get\_wordM()

{

FILE \*fp;

int line\_number = 0;

int i;

char current\_word[1000];

fp = fopen("medium.txt", "r");

if (fp == NULL)

{

perror("Error in opening file");

}

while (fgets(current\_word, 50, fp) != NULL)

{

line\_number++;

}

random\_number = rand() % line\_number;

rewind(fp);

for (line\_number = 0; line\_number != random\_number; line\_number++)

{

fgets(current\_word, 50, fp);

}

strcpy(word, current\_word);

fclose(fp);

}

void get\_wordH()

{

FILE \*fp;

int line\_number = 0;

int i;

char current\_word[1000];

fp = fopen("hard.txt", "r");

if (fp == NULL)

{

perror("Error in opening file");

}

while (fgets(current\_word, 50, fp) != NULL)

{

line\_number++;

}

random\_number = rand() % line\_number;

rewind(fp);

for (line\_number = 0; line\_number != random\_number; line\_number++)

{

fgets(current\_word, 50, fp);

}

strcpy(word, current\_word);

fclose(fp);

}

------0------