**CODE FOR HANG-MAN GAME USING C**

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <time.h>

#include <string.h>

#define WORDS 10

#define WORDLEN 40

#define CHANCE 6

bool srand\_called = false;

int i\_rnd(int i) {

if (!srand\_called) {

srand(time(NULL) << 10);

srand\_called = true;

}

return rand() % i;

}

char\* decrypt(char\* code) {

int hash = ((strlen(code) - 3) / 3) + 2;

char\* decrypt = malloc(hash);

char\* toFree = decrypt;

char\* word = code;

for (int ch = \*code; ch != '\0'; ch = \*(++code))

{

if((code - word + 2) % 3 == 1){

\*(decrypt++) = ch - (word - code + 1) - hash;

}

}

\*decrypt = '\0';

return toFree;

}

void printBody(int mistakes, char\* body) {

printf("\tMistakes :%d\n", mistakes);

switch(mistakes) {

case 6: body[6] = '\\'; break;

case 5: body[5] = '/'; break;

case 4: body[4] = '\\'; break;

case 3: body[3] = '|'; break;

case 2: body[2] = '/'; break;

case 1: body[1] = ')', body[0] = '('; break;

default: break;

}

printf("\t \_\_\_\_\_\_\_\_\_\n"

"\t| |\n"

"\t| %c %c\n"

"\t| %c %c %c\n"

"\t| %c %c\n"

"\t| \n"

"\t| ", body[0], body[1], body[2],

body[3], body[4], body[5], body[6]);

}

void printWord(char\* guess, int len) {

printf("\t");

for (int i = 0; i < len; ++i)

{

printf("%c ", guess[i]);

}

printf("\n\n");

}

int main() {

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

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

printf("\n\t - Maximum 6 mistakes are allowed.");

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

printf("\n\t - All words are name of very popular Websites. eg. Google");

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

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

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

char values[WORDS][WORDLEN] = {"N~mqOlJ^tZletXodeYgs","gCnDIfFQe^CdP^^B{hZpeLA^hv","7urtrtwQv{dt`>^}FaR]i]XUug^GI",

"aSwfXsxOsWAlXScVQmjAWJG","cruD=idduvUdr=gmcauCmg]","BQt`zncypFVjvIaTl]u=\_?Aa}F",

"iLvkKdT`yu~mWj[^gcO|","jSiLyzJ=vPmnv^`N]^>ViAC^z\_","xo|RqqhO|nNstjmzfiuoiFfhwtdh~","OHkttvxdp|[nnW]Drgaomdq"};

char \*body = malloc(CHANCE+1);

int id = i\_rnd(WORDS);

char \*word = decrypt(values[id]);

int len = strlen(word);

char \*guessed = malloc(len);

char falseWord[CHANCE];

memset(body, ' ', CHANCE+1);

memset(guessed, '\_', len);

char guess;

bool found;

char\* win;

int mistakes = 0;

setvbuf(stdin, NULL, \_IONBF, 0);

do {

found = false;

printf("\n\n");

printBody(mistakes, body);

printf("\n\n");

printf("\tFalse Letters : ");

if(mistakes == 0) printf("None\n");

for (int i = 0; i < mistakes; ++i)

{

printf("%c", falseWord[i]);

}

printf("\n\n");

printWord(guessed, len);

printf("\tGive me a alphabet in lower case : ");

do {scanf("%c",&guess);} while ( getchar() != '\n' );

for (int i = 0; i < len; ++i)

{

if(word[i] == guess) {

found = true;

guessed[i] = guess;

}

}

if(!found) {

falseWord[mistakes] = guess;

mistakes += 1;

}

win = strchr(guessed, '\_');

}while(mistakes < CHANCE && win != NULL);

if(win == NULL) {

printf("\n");

printWord(guessed, len);

printf("\n\tCongrats! You have won : %s\n\n", word);

} else {

printf("\n");

printBody(mistakes, body);

printf("\n\n\tBetter try next time. Word was %s\n\n", word);

}

free(body);

free(word);

free(guessed);

return EXIT\_SUCCESS;

}