

Relazione di Progetto

Programmazione Logica e Funzionale

Remedi Tommaso - 300535

Monaco Riccardo - 300537

Università degli Studi di Urbino Carlo Bo
Insegnamento di Programmazione Logica e Funzionale

1 Specifica del Problema

Scrivere un programma Haskell e un programma Prolog che permettono di giocare al gioco dell'impiccato (*Hangman Game*) da linea di comando. I programmi devono acquisire da tastiera la lettera che il giocatore pensa appartenga alla parola, composta esclusivamente da lettere minuscole, poi stampare a schermo lo stato dell'*omino*, la parola da indovinare con un "-" in corrispondenza delle lettere non ancora indovinate, e le lettere stesse altrimenti. La parola, partita per partita, viene scelta randomicamente da una lista di esse. Il giocatore inoltre avrà cinque possibilità di sbagliare prima di perdere.

2 Analisi del Problema

2.1 Dati di Ingresso del Problema

L'unico dato di ingresso del problema è il carattere digitato dall'utente.

2.2 Dati di Uscita del Problema

I dati di uscita del problema sono:

1. Una lista di lettere rappresentanti quelle che l'utente ha indovinato.
2. Un numero intero rappresentante i tentativi rimasti.
3. Una stringa di simboli rappresentante lo stato grafico dell'*omino*.

2.3 Relazioni Intercorrenti tra i Dati del Problema

La visualizzazione iniziale della parola da indovinare consiste in una serie di "-", uno per ogni lettera. Le lettere indovinate vengono, progressivamente col gioco, sostituite ai "-" della visualizzazione iniziale.

3 Progettazione dell'Algoritmo

3.1 Scelte di Progetto

Si è scelto di utilizzare una lista di stringhe per rappresentare le possibili parole da indovinare ed una lista di caratteri per le lettere indovinate.

La classica rappresentazione grafica dell'*omino* è divisa in due sezioni:

1. Una sezione comune, la parte alta e bassa del disegno, consistente in una stampa di caratteri in *ASCII Art*.
2. Una sezione dinamica, che cambia in base allo stato del gioco, anch'essa composta da caratteri in *ASCII Art*.

Vengono inoltre gestiti i seguenti casi:

- L'acquisizione dell'input dell'utente
- Il controllo sull'input
 - Numeri non ammessi
 - Maiuscole non ammesse
 - Eventuali caratteri oltre il primo scartati
- Il controllo dell'eventuale corrispondenza dell'input con una delle lettere della parola da indovinare
- La visualizzazione dello stato corrente della parola da indovinare
- La scelta della parola da indovinare
- La diminuzione dei tentativi disponibili
- Il caso di vittoria
- Il caso di sconfitta
- Il progresso del gioco

3.2 Passi dell'Algoritmo

I passi dell'algoritmo per risolvere il problema sono i seguenti:

1. Selezionare la parola da indovinare
2. Inizializzare il numero di tentativi disponibili
3. Acquisire in input un carattere digitato dall'utente
4. Controllare la validità dell'input
5. Controllare se l'input corrisponde ad una lettera presente nella parola da indovinare
 - Se la lettera inserita dall'utente è contenuta nella parola da indovinare, la si aggiunge alla lista di lettere indovinate.
 - Se la lettera inserita dall'utente non è contenuta nella parola da indovinare, si decrementa il numero di tentativi disponibili.

6. Controllare se sono stati esauriti i tentativi disponibili.
7. Se i tentativi sono esauriti visualizzare il messaggio di sconfitta e la parola completa.
8. Se ci sono ancora tentativi, controllare se la parola è stata indovinata.
9. Se la parola è stata indovinata, visualizzare il messaggio di vittoria e la parola completa.
10. Se ci sono ancora tentativi e la parola non è stata indovinata, visualizzazione dell'*omino* e della parola "bucata".
11. Ritorno al passo di acquisizione dell'input. (passo 3)

4 Implementazione dell'Algoritmo

File sorgente hangman_game.hs

```
{- Logical and Functional Programming Project a.y. 2023-2024
   Lecturer: Prof. Marco Bernardo
   Students:
       Tommaso Remedi - 300535
       Riccardo Monaco - 300537 -}
```



```
{- Haskell program to play the hangman game. -}
```



```
import System.Random -- needed to randomly extract a number as the index of the word to guess
import Data.List -- needed to use nub, which removes duplicate items from a list
import System.IO -- needed to acquire the user input
```



```
main :: IO ()
main = envSetup 6
```



```
{- Words to guess list. -}
```



```
wordsToGuess :: [String]
wordsToGuess = ["haskell", "programming", "functional", "language", "computation"]
```



```
{- The function envSetup sets the game environment, picking the word to guess. -}
```



```
envSetup :: Int -> IO ()
envSetup remainingAttempts = do
    putStrLn "Welcome to the Hangman Game!"
    wordToGuess <- selectWord
    play wordToGuess [] remainingAttempts
```



```
{- The function selectWord randomly selects a word from the given list: -}
```



```
selectWord :: IO String
selectWord = do
    index <- randomRIO (0, length wordsToGuess - 1)
    return $ wordsToGuess !! index
```



```
{- The function play manages the game, updates the guessed letters and the attempts:
   - The first parameter stands for the word to guess;
   - The second parameter stands for the letters already guessed;
   - The third parameter stands for the remaining attempts. -}
```



```
play :: String -> String -> Int -> IO ()
play wordToGuess guessedLetters remainingAttempts
    | checkGuessed wordToGuess guessedLetters = handleWin wordToGuess
    | remainingAttempts == 0 = handleLoss wordToGuess
    | otherwise = handleInProgress wordToGuess guessedLetters remainingAttempts
```



```
{- The function handleLoss handles the case when the user runs out of attempts:
```

```

    - The first parameter stands for the word to guess. -}

handleLoss :: String -> IO ()
handleLoss wordToGuess = do

    putStr (drawHangman 0)
    putStrLn "You've Lost! Your hangman has been hanged."
    putStrLn $ "The secret word was: " ++ wordToGuess

{- The function handleWin handles the case when the user wins:
    - The first parameter stands for the word to guess. -}

handleWin :: String -> IO ()
handleWin wordToGuess = do
    putStrLn $ "You've guessed it! The secret word was: " ++ wordToGuess

{- The function handleInProgress handles the in-progress scenario:
    - The first parameter stands for the word to guess;
    - The second parameter stands for the letters already guessed;
    - The third parameter stands for the remaining attempts. -}

handleInProgress :: String -> String -> Int -> IO ()
handleInProgress wordToGuess guessedLetters remainingAttempts = do
    putStrLn $ "Current word: " ++ renderWord wordToGuess guessedLetters
    putStr (drawHangman remainingAttempts)
    putStrLn $ "Remaining attempts: " ++ show remainingAttempts
    insertedLetter <- readFirstChar
    updateAttempt wordToGuess guessedLetters insertedLetter remainingAttempts

{- The function readFirstChar reads and checks the validity of the input:
    - The first parameter stands for the first inserted char. -}

readFirstChar :: IO Char
readFirstChar = do
    putStr "Guess a letter: "
    hFlush stdout
    char <- getChar
    _ <- getLine
    if char `elem` ['a'..'z']
        then return char
        else do
            putStrLn "Please enter a valid lowercase letter."
            readFirstChar

{- The function updateAttempt updates the guessed letters list with eventually a new one:
    - The first parameter stands for the word to guess;
    - The second parameter stands for the letters already guessed;
    - The third parameter stands for the letter the user has inserted;
    - The fourth parameter stands for the remaining attempts. -}

updateAttempt :: String -> String -> Char -> Int -> IO ()

```

```

updateAttempt wordToGuess guessedLetters insertedLetter remainingAttempts
  | insertedLetter `elem` guessedLetters = do
    putStrLn "You've already guessed this letter!"
    play wordToGuess guessedLetters remainingAttempts
  | insertedLetter `elem` wordToGuess = do
    putStrLn "Letter is in the word!"
    play wordToGuess (nub $ guessedLetters ++ [insertedLetter]) remainingAttempts
  | otherwise = do
    putStrLn "Wrong letter!"
    play wordToGuess guessedLetters (remainingAttempts - 1)

{- The function checkGuessed returns true if all the letters have been guessed:
- The first parameter stands for the word to guess;
- The second parameter stands for the letters already guessed. -}

checkGuessed :: String -> String -> Bool
checkGuessed wordToGuess guessedLetters =
  all (`elem` guessedLetters) wordToGuess

{- The function renderWord prints a letter if it is found in the word,
an underscore if not:
- The first parameter stands for the letters of the word to guess;
- The second parameter stands for the letters already guessed. -}

renderWord :: String -> String -> String
renderWord wordToGuess guessedLetters =
  [if c `elem` guessedLetters || c == ' ' then c else '_' | c <- wordToGuess]

{- The function drawHangman prints the characters to draw the countours of the hangman:
- The first parameter stands for the remaining attempts. -}

drawHangman :: Int -> String
drawHangman remainingAttempts =
  unlines $
    [ " +---+"
    , " |   |"
    ] ++
    drawCase remainingAttempts ++
    ["====="]
  where
    drawCase 6 = ["      |", "      |", "      |"]
    drawCase 5 = [" 0  |", "      |", "      |"]
    drawCase 4 = [" 0  |", "      |", "      |"]
    drawCase 3 = [" 0  |", " /|  |", "      |"]
    drawCase 2 = [" 0  |", " /|\ |", "      |"]
    drawCase 1 = [" 0  |", " /|\ |", " /   |"]
    drawCase 0 = [" 0  |", " /|\ |", " / \  |"]
    drawCase _ = []

```

File sorgente hangman_game.pl

```
/* Logical and Functional Programming Project a.y. 2023-2024
   Lecturer: Prof. Marco Bernardo
   Students:
       Tommaso Remedi - 300535
       Riccardo Monaco - 300537 */

/* Prolog program to play the hangman game. */

main :-
    env_setup(6).

/* Words to guess list. */

words_list(['haskell', 'programming', 'functional', 'language', 'computation']).

/* The predicate env_setup sets the game environment, picking the word to guess:
   - The first parameter stands for the total attempts. */

env_setup(Remaining_Attempts) :-
    clean_console,
    write('Welcome to the Hangman Game!'), nl,
    sel_word(Word_To_Guess),
    play(Word_To_Guess, [], Remaining_Attempts).

/* The predicate sel_world randomly selects a word from the given list:
   - The first parameter stands for the word to be selected. */

sel_word(Word) :-
    words_list(Words_List),
    length(Words_List, List_Length),
    random(0, List_Length, Word_Index),
    nth0(Word_Index, Words_List, Atom_Word),
    atom_chars(Atom_Word, Word).

/* The predicate play manages the game, it updates the guessed letters and the attempts:
   - The first parameter stands for the word to guess;
   - The second parameter stands for the letters already guessed;
   - The third parameter stands for the remaining attempts. */

play(Word_To_Guess, Guessed_Letters, Remaining_Attempts) :-
    Remaining_Attempts == 0 ->
        handle_loss(Word_To_Guess)
    ;
    check_guessed(Word_To_Guess, Guessed_Letters) ->
        handle_win(Word_To_Guess)
    ;
        handle_in_progress(Word_To_Guess, Guessed_Letters, Remaining_Attempts).

/* The predicate handle_loss handles the case when the user runs out of attempts:
```



```

- The first parameter stands for the word to guess. */

handle_loss(Word_To_Guess) :-
    clean_console,
    draw_hangman(0),
    write('You\'ve Lost!'), nl,
    write('The word to guess was: '), print_list(Word_To_Guess), nl.

/* The predicate handle_win handles the case when the user wins:
- The first parameter stands for the word to guess. */

handle_win(Word_To_Guess) :-
    clean_console,
    write('You\'ve won! The secret word was: '), print_list(Word_To_Guess), nl.

/* The predicate handle_in_progress handles the in-progress scenario:
- The first parameter stands for the word to guess;
- The second parameter stands for the letters already guessed;
- The third parameter stands for the remaining attempts. */

handle_in_progress(Word_To_Guess, Guessed_Letters, Remaining_Attempts) :-
    clean_console,
    write('Actual word:'), nl,
    render_word(Word_To_Guess, Guessed_Letters), nl, nl,
    draw_hangman(Remaining_Attempts),
    write('Remaining attempts: '), write(Remaining_Attempts), nl, nl,
    write('Guess a Letter: '), nl,
    read_first_char(Inserted_Letter),
    upd_attempt(Word_To_Guess, Guessed_Letters, Inserted_Letter, Remaining_Attempts).

/* The predicate read_first_char reads and checks the validity of the input:
- The first parameter stands for the first inserted char. */

read_first_char(First_Char) :-
    get_code(First_Code),
    atom_codes(First_Char, [First_Code]),
    (
        (First_Code >= 97, First_Code =< 122) ->
            read_remaining_chars(First_Code, Remaining_Codes)
        ;
        write('Please, enter a lowercase letter. '), nl,
        read_remaining_chars(First_Code, Remaining_Codes),
        clean_console,
        write('Guess a Letter: '), nl,
        read_first_char(_)
    ).

/* The predicate read_remaining_chars cleans up any leftover of the input:
- The first parameter stands for the first code read;

```

```

- The second parameters stands for the eventual other codes inserted. */

read_remaining_chars(10, []) :- !.
read_remaining_chars(Code, [Code | Remaining_Codes]) :-
    get_code(New_Code),
    read_remaining_chars(New_Code, Remaining_Codes).

/* The predicate upd_attempt updates the guessed letters list with eventually a new one:
- The first parameter stands for the word to guess;
- The second parameter stands for the letters already guessed;
- The third parameter stands for the letter the user has inserted;
- The fourth parameter stands for the remaining attempts. */

upd_attempt(Word_To_Guess, Guessed_Letters, Inserted_Letter, Remaining_Attempts) :-
    write('_____'), nl, nl,
    memberchk(Inserted_Letter, Guessed_Letters) ->
        write('Already guessed letter!'), nl,
        play(Word_To_Guess, Guessed_Letters, Remaining_Attempts)
    ;
    memberchk(Inserted_Letter, Word_To_Guess) ->
        write('Letter is in the word!'), nl,
        append(Guessed_Letters, [Inserted_Letter], Upd_Guessed_Letters),
        play(Word_To_Guess, Upd_Guessed_Letters, Remaining_Attempts)
    ;
    write('Wrong letter!'), nl,
    Upd_Remaining_Attempts is Remaining_Attempts - 1,
    play(Word_To_Guess, Guessed_Letters, Upd_Remaining_Attempts).

/* The predicate render_word prints a letter if it is found in the word, a "_" if not:
- The first parameter stands for the letters of the word to guess;
- The second parameter stands for the letters already guessed. */

render_word([], _).
render_word([C|Word_To_Guess], Guessed_Letters) :-
    (
        memberchk(C, Guessed_Letters) ->
            write(C), write(' ')
        ;
        write('_ ')
    ),
    render_word(Word_To_Guess, Guessed_Letters).

/* The predicate check_guessed returns true if all the letters have been guessed:
- The first parameter stands for the word to guess;
- The second parameter stands for the letters already guessed. */

check_guessed(Word_To_Guess, Guessed_Letters) :-
    subtract(Word_To_Guess, Guessed_Letters, []).

```

```

/* The predicate draw_hangman prints the characters to draw the countours of the hangman:
   - The first parameter stands for the remaining attempts. */

draw_hangman(Remaining_Attempts) :-
    write(' +---+'), nl,
    write(' |   |'), nl,
    draw_case(Remaining_Attempts),
    write('====='), nl.

/* The auxiliary predicate draw_case draws the specific "state" of the hangman:
   - The first parameter stands for the reached level of error. */

draw_case(6) :-
    write('      |'), nl,
    write('      |'), nl,
    write('      |'), nl.
draw_case(5) :-
    write(' 0   |'), nl,
    write('      |'), nl,
    write('      |'), nl.
draw_case(4) :-
    write(' 0   |'), nl,
    write(' |   |'), nl,
    write('      |'), nl.
draw_case(3) :-
    write(' 0   |'), nl,
    write(' /|  |'), nl,
    write('      |'), nl.
draw_case(2) :-
    write(' 0   |'), nl,
    write(' /|\ |'), nl,
    write('      |'), nl.
draw_case(1) :-
    write(' 0   |'), nl,
    write(' /|\ |'), nl,
    write(' /   |'), nl.
draw_case(0) :-
    write(' 0   |'), nl,
    write(' /|\ |'), nl,
    write(' / \  |'), nl.

/* The predicate clean_console prints a new line. */

clean_console :-
    nl.

/* The predicate print_list pretty prints a list:
   - The first parameter stands for the list to print. */

print_list([]).
print_list([X|Xs]) :-

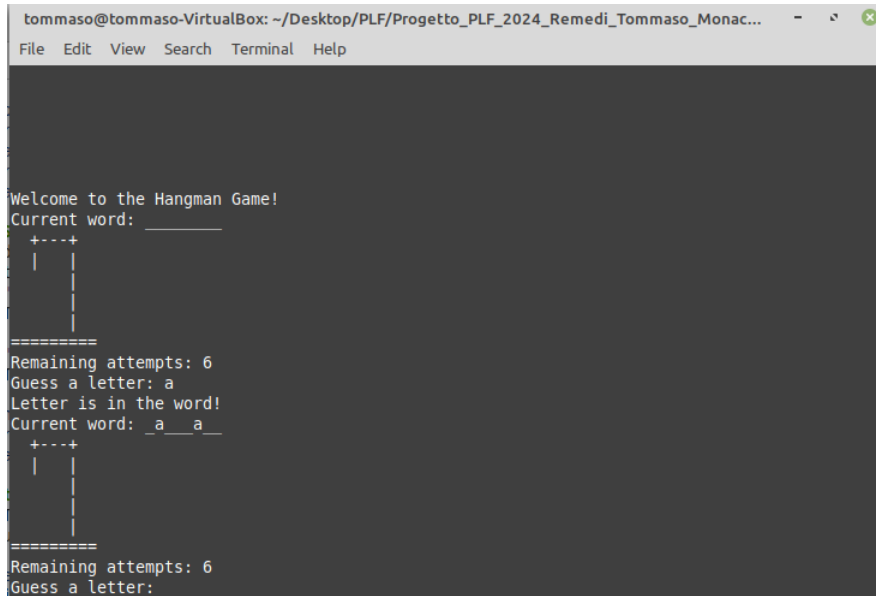
```

```
write(X),  
print_list(Xs).
```

5 Testing del Programma

Test Haskell 1

Inserimento lettera corretta



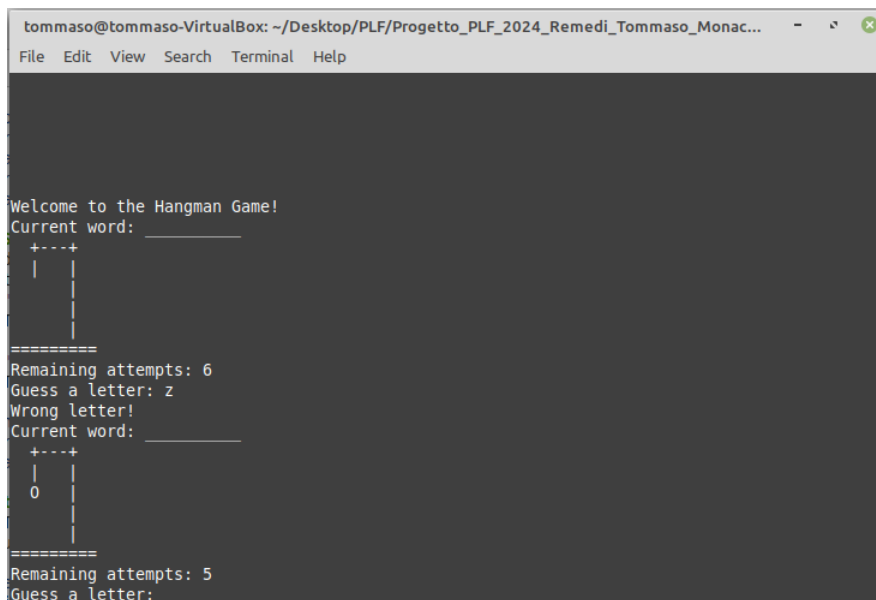
```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monac...  
File Edit View Search Terminal Help  
  
Welcome to the Hangman Game!  
Current word: _____  
+---+  
|  
|  
|  
+---+  
=====
```

Remaining attempts: 6
Guess a letter: a
Letter is in the word!
Current word: _a_a_
+---+
|
|
|
+---+
=====

Remaining attempts: 6
Guess a letter:

Test Haskell 2

Inserimento lettera sbagliata



```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monac...  
File Edit View Search Terminal Help  
  
Welcome to the Hangman Game!  
Current word: _____  
+---+  
|  
|  
|  
+---+  
=====
```

Remaining attempts: 6
Guess a letter: z
Wrong letter!
Current word: _____
+---+
|
0
|
|
+---+
=====

Remaining attempts: 5
Guess a letter:

Test Haskell 3

Inserimento numero

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monac...  
File Edit View Search Terminal Help  
  
Welcome to the Hangman Game!  
Current word: _____  
+---+  
|  
|  
|  
|  
+---+  
=====
```

Remaining attempts: 6
Guess a letter: 7
Please enter a valid lowercase letter.
Guess a letter:

Test Haskell 4

Inserimento lettera maiuscola

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monac...  
File Edit View Search Terminal Help  
  
Welcome to the Hangman Game!  
Current word: _____  
+---+  
|  
|  
|  
|  
+---+  
=====
```

Remaining attempts: 6
Guess a letter: C
Please enter a valid lowercase letter.
Guess a letter:

Test Haskell 5

Inserimento carattere non alfanumerico

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|
|
|
|
=====
Remaining attempts: 6
Guess a letter: @
Please enter a valid lowercase letter.
Guess a letter: #
Please enter a valid lowercase letter.
Guess a letter: (
Please enter a valid lowercase letter.
Guess a letter: 
```

Test Haskell 6

Inserimento stessa lettera due volte

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|
|
|
|
=====
Remaining attempts: 6
Guess a letter: a
Letter is in the word!
Current word: _____a_
+---+
|
|
|
|
=====
Remaining attempts: 6
Guess a letter: a
You've already guessed this letter!
Current word: _____a_
+---+
|
|
|
|
=====
Remaining attempts: 6
Guess a letter: 
```

Test Haskell 7

Vittoria

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|   |
|   |
|   |
=====
Remaining attempts: 6
Guess a letter: h
Wrong letter!
Current word: _____
+---+
|   |
0   |
|   |
|   |
=====
Remaining attempts: 5
Guess a letter: e
Letter is in the word!
Current word: _____e
+---+
|   |
0   |
|   |
|   |
=====
Remaining attempts: 5
Guess a letter: l
Letter is in the word!
Current word: l_____e
+---+
|   |
0   |
|   |
|   |
=====
Remaining attempts: 5
Guess a letter: a
Letter is in the word!
Current word: la__a_e
+---+
|   |
0   |
|   |
|   |
```



```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help

| |
0 |
|
=====
Remaining attempts: 5
Guess a letter: l
Letter is in the word!
Current word: l____e
+---+
| |
0 |
|
=====
Remaining attempts: 5
Guess a letter: a
Letter is in the word!
Current word: la__a_e
+---+
| |
0 |
|
=====
Remaining attempts: 5
Guess a letter: n
Letter is in the word!
Current word: lan_a_e
+---+
| |
0 |
|
=====
Remaining attempts: 5
Guess a letter: g
Letter is in the word!
Current word: lang_age
+---+
| |
0 |
|
=====
Remaining attempts: 5
Guess a letter: u
Letter is in the word!
You've guessed it! The secret word was: language
```

Test Haskell 8

Sconfitta

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6
Guess a letter: z
Wrong letter!
Current word: _____
+---+
|   |
| 0  |
|   |
+---+

=====
Remaining attempts: 5
Guess a letter: d
Wrong letter!
Current word: _____
+---+
|   |
| 0  |
|   |
+---+

=====
Remaining attempts: 4
Guess a letter: h
Wrong letter!
Current word: _____
+---+
|   |
| 0  |
| /  |
+---+

=====
Remaining attempts: 3
Guess a letter: y
Wrong letter!
Current word: _____
+---+
|   |
| 0  |
| /\ |
+---+

=====
Remaining attempts: 2
```

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help
Wrong letter!
Current word: _____
+---+
|   |
0   |
|   |
+---+
=====
Remaining attempts: 4
Guess a letter: h
Wrong letter!
Current word: _____
+---+
|   |
0   |
/|  |
+---+
=====
Remaining attempts: 3
Guess a letter: y
Wrong letter!
Current word: _____
+---+
|   |
0   |
/|\ |
+---+
=====
Remaining attempts: 2
Guess a letter: w
Wrong letter!
Current word: _____
+---+
|   |
0   |
/|\ |
/   |
+---+
=====
Remaining attempts: 1
Guess a letter: q
Wrong letter!
+---+
|   |
0   |
/|\ |
/ \ |
+---+
=====
You've Lost! Your hangman has been hanged.
The secret word was: computation
```

Test Haskell 9

Sbagliare fino ad avere un tentativo rimasto, poi vittoria

```
tommaso@tommaso-VirtualBox
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: z
Wrong letter!
Current word: _____
+---+
|
| 0
|
|
|
=====
Remaining attempts: 5
Guess a letter: x
Wrong letter!
Current word: _____
+---+
|
| 0
|
|
|
=====
Remaining attempts: 4
Guess a letter: y
Wrong letter!
Current word: _____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 3
Guess a letter: q
Wrong letter!
Current word: _____
+---+
|
| 0
|
| / \
|
|

tommaso@tommaso-VirtualB
File Edit View Search Terminal Help

Wrong letter!
Current word: _____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 2
Guess a letter: z
Wrong letter!
Current word: _____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 1
Guess a letter: h
Letter is in the word!
Current word: h_____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 1
Guess a letter: a
Letter is in the word!
Current word: ha_____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 1
Guess a letter: s
Letter is in the word!
Current word: has_____
+---+
|
| 0
|
| / \
|
|
=====
Remaining attempts: 1
Guess a letter: k
```

```
tommaso@tommaso-VirtualBox: ~/Desktop/PLF/Progetto_PLF_2024_Remedi_Tommaso_Monaco_Riccardo
File Edit View Search Terminal Help
|
0 |
/|\ |
/ |
=====
Remaining attempts: 1
Guess a letter: a
Letter is in the word!
Current word: ha__
+---+
|
0 |
/|\ |
/ |
=====
Remaining attempts: 1
Guess a letter: s
Letter is in the word!
Current word: has__
+---+
|
0 |
/|\ |
/ |
=====
Remaining attempts: 1
Guess a letter: k
Letter is in the word!
Current word: hask__
+---+
|
0 |
/|\ |
/ |
=====
Remaining attempts: 1
Guess a letter: e
Letter is in the word!
Current word: haske__
+---+
|
0 |
/|\ |
/ |
=====
Remaining attempts: 1
Guess a letter: l
Letter is in the word!
You've guessed it! The secret word was: haskell
```

Test Haskell 10

Indovinare tutte le lettere a parte una, poi sconfitta

```
tommaso@b
File Edit View Search Terminal Help

Welcome to the Hangman Game!
Current word: _____
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: p
Letter is in the word!
Current word: _p_____
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: c
Letter is in the word!
Current word: c_p_____
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: o
Letter is in the word!
Current word: co_p_____o_
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: m
Letter is in the word!
Current word: comp_____o_
+---+
|
|
|
=====
```

```
tommaso@tommaso-VirtualBox: ~/Desk
File Edit View Search Terminal Help

=====
Remaining attempts: 6
Guess a letter: u
Letter is in the word!
Current word: compu__o_
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: t
Letter is in the word!
Current word: comput_t_o_
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: a
Letter is in the word!
Current word: computat_o_
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: i
Letter is in the word!
Current word: computatio_
+---+
|
|
|
=====
Remaining attempts: 6
Guess a letter: p
You've already guessed this letter!
Current word: computatio_
+---+
|
|
|
=====
```

```

tommaso@tommaso-VirtualBox: ~/Des
File Edit View Search Terminal Help
=====
Remaining attempts: 6
Guess a letter: p
You've already guessed this letter!
Current word: computatio_
+---+
|   |
|   |
|   |
=====
Remaining attempts: 6
Guess a letter: q
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
|   |
|   |
=====
Remaining attempts: 5
Guess a letter: r
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
|   |
|   |
=====
Remaining attempts: 4
Guess a letter: x
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
/ \ |
|   |
=====
Remaining attempts: 3
Guess a letter: z
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
/ \ |
|   |
=====

```

```

tommaso@tommaso-VirtualBox: ~/Deskt
File Edit View Search Terminal Help
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
|   |
=====
Remaining attempts: 4
Guess a letter: x
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
/ \ |
|   |
=====
Remaining attempts: 3
Guess a letter: z
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
/ \ |
|   |
=====
Remaining attempts: 2
Guess a letter: v
Wrong letter!
Current word: computatio_
+---+
|   |
0   |
/ \ |
/   |
=====
Remaining attempts: 1
Guess a letter: k
Wrong letter!
+---+
|   |
0   |
/ \ |
/   |
=====
You've Lost! Your hangman has been hanged.
The secret word was: computation

```

Test Prolog 1

Inserimento lettera corretta

```
| ?- main.

Welcome to the Hangman Game!

Actual word:
- - - - -
+---+
|   |
|   |
|   |
=====
Remaining attempts: 6

Guess a Letter:
a
_____

Letter is in the word!

Actual word:
_ a _ _ _ _ _
+---+
|   |
|   |
|   |
=====
Remaining attempts: 6

Guess a Letter:
```

Test Prolog 2

Inserimento lettera sbagliata

```
| ?- main.

Welcome to the Hangman Game!

Actual word:
- - - - -
+---+
|   |
|   |
|   |
=====
Remaining attempts: 6

Guess a Letter:
z
_____

Wrong letter!

Actual word:
- - - - -
+---+
|   |
| 0 |
|   |
|   |
=====
Remaining attempts: 5

Guess a Letter:
```


Test Prolog 3

Inserimento numero

```
| ?- main.  
Welcome to the Hangman Game!  
Actual word:  
-----  
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
5
Please, enter a lowercase letter.

Guess a Letter:

Test Prolog 4

Inserimento lettera maiuscola

```
| ?- main.  
Welcome to the Hangman Game!  
Actual word:  
-----  
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
F
Please, enter a lowercase letter.

Guess a Letter:

Test Prolog 5

Inserimento carattere non alfanumerico

```
| ?- main.  
  
Welcome to the Hangman Game!  
  
Actual word:  
-----  
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
@
Please, enter a lowercase letter.
Guess a Letter:

Test Prolog 6

Inserimento stessa lettera due volte

```
Welcome to the Hangman Game!  
  
Actual word:  
-----  
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
a

Letter is in the word!

Actual word:
_ a _ _ _ _ _

```
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
a

Already guessed letter!

Actual word:
_ a _ _ _ _ _

```
+---+  
|   |  
|   |  
|   |  
+---+  
=====
```

Remaining attempts: 6

Guess a Letter:
|

Test Prolog 7

Vittoria

```
/home/tommaso/Desktop/PLF/Progetto_PLF_2024_Remedi_7
17 ms

(2 ms) yes
| ?- main.

Welcome to the Hangman Game!

Actual word:
_ _ _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
p
_____

Letter is in the word!

Actual word:
_ _ _ p _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
c
_____

Letter is in the word!

Actual word:
c _ _ p _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
o
_____

Letter is in the word!

Actual word:
c o _ p _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
m
_____

Letter is in the word!

Actual word:
c o m p _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
u
_____

Letter is in the word!
```

```
c _ _ p _ _ _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
o
_____

Letter is in the word!

Actual word:
c o _ p _ _ _ o _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
m
_____

Letter is in the word!

Actual word:
c o m p _ _ _ o _

+---+
|   |
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
u
_____

Letter is in the word!
```


Test Prolog 8

Sconfitta

```
| ?- main.  
Welcome to the Hangman Game!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
|   |  
+---+  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
y
```

```
Wrong letter!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
0   |  
|   |  
+---+  
=====
```

```
Remaining attempts: 5
```

```
Guess a Letter:
```

```
z
```

```
Wrong letter!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
0   |  
|   |  
+---+  
=====
```

```
Remaining attempts: 4
```

```
Guess a Letter:
```

```
Guess a Letter:
```

```
x
```

```
Wrong letter!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
0   |  
/|  |  
|   |  
+---+  
=====
```

```
Remaining attempts: 3
```

```
Guess a Letter:
```

```
q
```

```
Wrong letter!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
0   |  
/|\  |  
|   |  
+---+  
=====
```

```
Remaining attempts: 2
```

```
Guess a Letter:
```

```
k
```

```
Wrong letter!
```

```
Actual word:
```

```
-----  
+---+  
|   |  
0   |  
/|\  |  
/   |  
+---+  
=====
```

```
q
Wrong letter!
Actual word:
-----
+---+
|   |
0   |
/|\  |
=====
Remaining attempts: 2
Guess a Letter:
k
Wrong letter!
Actual word:
-----
+---+
|   |
0   |
/|\  |
=====
Remaining attempts: 1
Guess a Letter:
y
Wrong letter!
+---+
|   |
0   |
/|\  |
=====
You've Lost!
The word to guess was: programming
yes
```

Test Prolog 9

Sbagliare fino ad avere un tentativo rimasto, poi vittoria

```
| ?- main.  
Welcome to the Hangman Game!
```

Actual word:

A diagram of a rectangular field. A dashed line runs horizontally across the middle of the field. Two points, marked with '+', are located on the dashed line. A vertical dashed line extends from the right point down to the bottom edge of the field.

Remaining attempts: 6

Guess a Letter:
q

Wrong letter!

```
Actual word:
-----
+---+
|   |
0   |
```

Remaining attempts: 5

Guess a Letter:
W

Wrong letter!

Actual word:

$$\begin{array}{c} + \quad - \quad - \quad + \\ | \qquad | \\ 0 \qquad | \\ | \qquad | \end{array}$$

Remaining attempts: 4

Guess a Letter:

```
=====
Remaining attempts: 4
```

Guess a Letter:
x

Wrong letter!

Actual word:

$$\begin{array}{c} + \quad - \quad + \\ | \quad \quad | \\ 0 \quad \quad | \\ / | \quad | \end{array}$$

Remaining attempts: 3

Guess a Letter:
Z

Wrong letter!

Actual word:

$$\begin{array}{c} + \quad - \quad - \quad + \\ | \qquad | \\ 0 \qquad | \\ / \backslash \quad | \end{array}$$

Remaining attempts: 2

Guess a Letter:
d

Wrong letter!

Actual word:

$$\begin{array}{c} + \quad - \quad - \quad + \\ | \quad \quad | \\ 0 \quad \quad | \end{array}$$

```

Wrong letter!
Actual word:
-----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
h

```

```

Letter is in the word!
Actual word:
h -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
a

```

```

Letter is in the word!
Actual word:
h a -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:

```

```

Letter is in the word!
Actual word:
h a -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
s

```

```

Letter is in the word!
Actual word:
h a s -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
k

```

```

Letter is in the word!
Actual word:
h a s k -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
l

```

```

+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
k

```

```

Letter is in the word!
Actual word:
h a s k -----
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
l

```

```

Letter is in the word!
Actual word:
h a s k _ l l
+---+
|   |
| 0 |
|/|\|
|   |
+---+
Remaining attempts: 1
Guess a Letter:
e

```

```

Letter is in the word!
You've won! The secret word was: haskell
(1 ms) yes

```


Test Prolog 10

Indovinare tutte le lettere a parte una, poi sconfitta

```
| ?- main.
```

```
Welcome to the Hangman Game!
```

```
Actual word:
```

```
- - - - -
```

```
  +---+  
  |   |  
  |   |  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
a
```

```
Letter is in the word!
```

```
Actual word:
```

```
_ a _ _ _ a _ _
```

```
  +---+  
  |   |  
  |   |  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
l
```

```
Letter is in the word!
```

```
Actual word:
```

```
l a _ _ _ a _ _
```

```
  +---+  
  |   |  
  |   |  
=====
```

```
Remaining attempts: 6
```

```
  |   |  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
n
```

```
Letter is in the word!
```

```
Actual word:
```

```
l a n _ _ a _ _
```

```
  +---+  
  |   |  
  |   |  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
g
```

```
Letter is in the word!
```

```
Actual word:
```

```
l a n g _ a g _
```

```
  +---+  
  |   |  
  |   |  
=====
```

```
Remaining attempts: 6
```

```
Guess a Letter:
```

```
u
```

```
Letter is in the word!
```

```
Actual word:
```

```
l a n g u a g _
```

```
  +---+  
  |   |  
  |   |
```

```

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 6

Guess a Letter:
w

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 5

Guess a Letter:
q

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 4

Guess a Letter:
z

Wrong letter!

```

```

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 3

Guess a Letter:
x

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 2

Guess a Letter:
c

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 1

Guess a Letter:

```

```

x

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 2

Guess a Letter:
c

Wrong letter!

Actual word:
l a n g u a g _

+---+
|   |
|   |
+---+

=====
Remaining attempts: 1

Guess a Letter:
v

Wrong letter!

+---+
|   |
|   |
+---+

=====
You've Lost!
The word to guess was: language
(1 ms) yes

```