

PROGETTO ESERCIZI VERBI IRREGOLARI

Descrizione del progetto

Si vuole realizzare un programma che ci aiuti a studiare l'inglese, in particolare i verbi irregolari. Vengono proposti all'utente 10 verbi.

La modalità di verifica può cambiare di volta in volta infatti può venir chiesto all'utente di scrivere il "Simple past" oppure il "Participio passato" o la traduzione partendo dalla forma base oppure di scrivere la forma base data la traduzione in italiano.

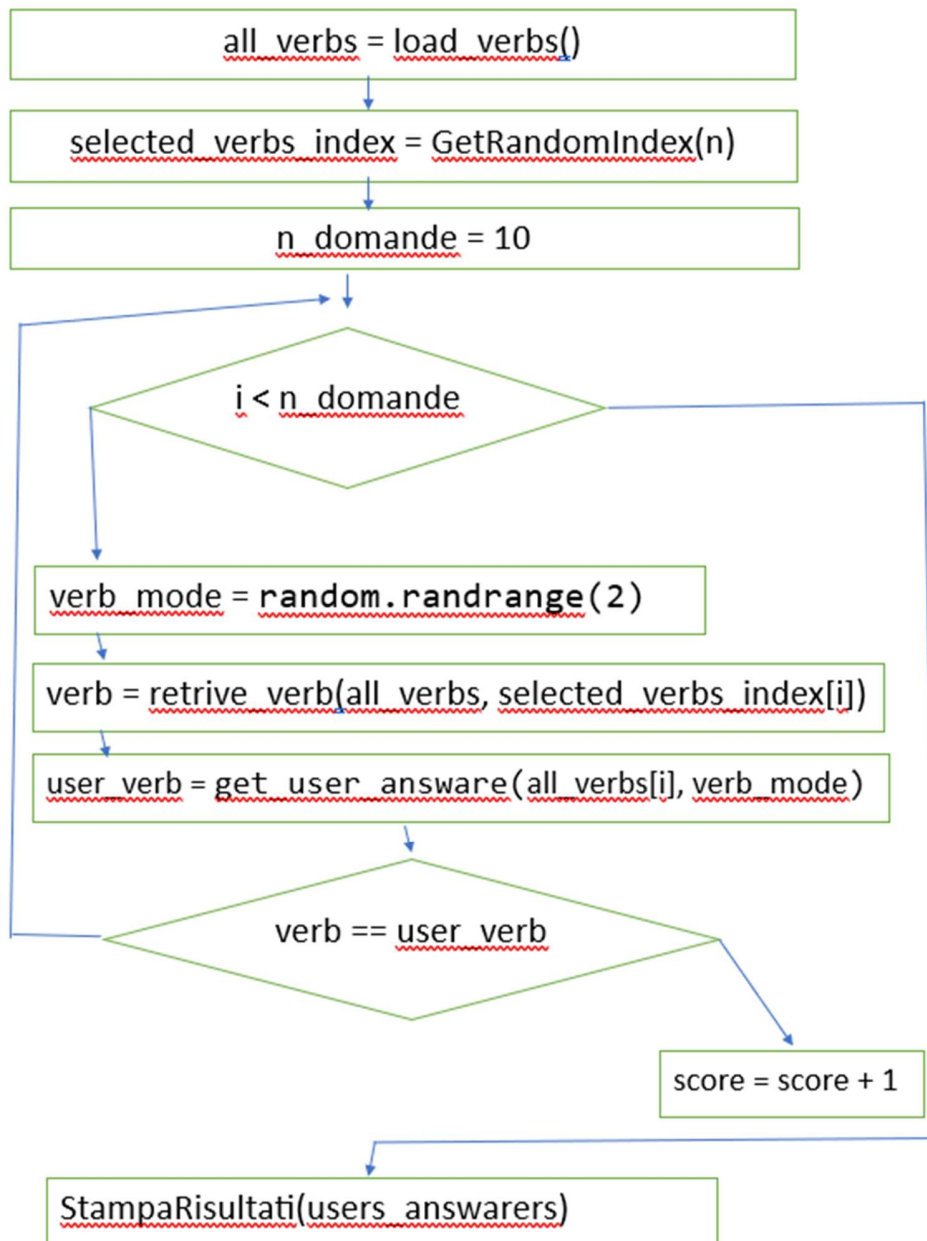
Una volta risposto a 10 verbi si visualizza il risultato e la relativa correzione.

Requisiti

- Il software deve permettere all'utente, se vuole, di poter ripetere il test senza dover uscire e rientrare nel programma.
- Il software permette di testare la propria competenza su tutti e tre i modi verbali.
- Permettere la stampa del risultato in formato "tabellare carino"
- Dare corretta come risposta sia "was" che "were"

System Overview

Il sistema nel suo complesso risulta così composto:



Inizialmente vengono caricati i verbi da indovinare in una lista. Il passo successivo è estrarre in modo casuale dieci verbi tra tutti quelli possibili.

Una volta conclusa la parte di setup viene chiesto un verbo alla volta all'utente. In modo random vengono chiesti "Simple past" oppure il "Participio passato" o la traduzione. Ovviamente in base al verbo bisogna stampare il corretto messaggio all'utente.

Una volta che l'utente ha dato la risposta la si confronta con la risposta corretta e si aggiorna il punteggio.

Una volta chiesti tutti i verbi si stampa la tabella con i risultati.

Le variabili globali di cui abbiamo bisogno sono:

all_verbs = lista che contiene tutti i verbi. La singola riga memorizza i tempi verbali separati dalla virgola: Do, Did, Done, Fare

selected_verbs = lista dei 10 verbi scelti per il test. È una lista che contiene gli indici dei verbi da recupera nella lista "all_verbs"

n_domande =

Tutti i verbi sono memorizzati direttamente in una singola stringa dove i verbi sono divisi dal carattere '|': Abide, Abode, Abode, Sopportare|

Per stampare la tabella ...

Funzioni specifiche

In questa parte della documentazione si trovano tutte le funzioni utilizzate nel progetto:

Funzione “get_random_index(n, max)”:

Per riferimento guardare libro pag. 130. Se applicabile

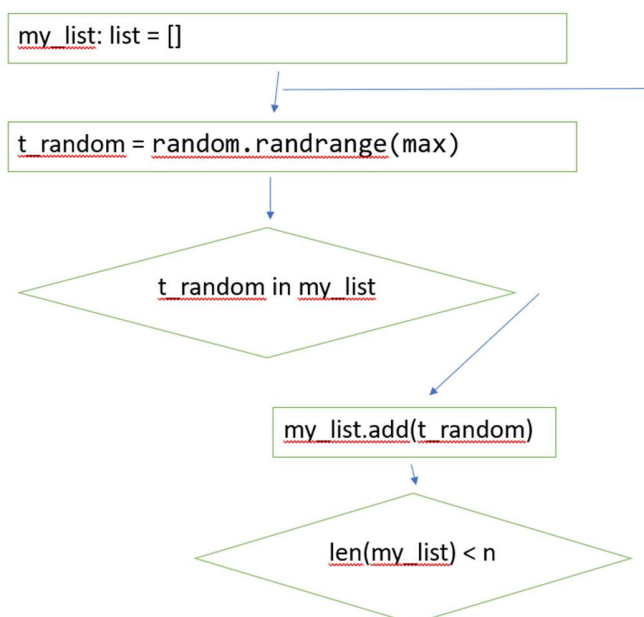
Questa funzione ci permette di estrarre i 10 numeri random associati alla lista dei verbi.

Input:

- n= numero degli elementi della lista di output
- max = numero di massimo estraibile

Output:

- my_list = lista di numeri random, ogni numero della lista deve essere univoco.



Funzione load_verbs();

- Descrizione
- Variabili
- Schema

Funzione retrieve_verb(current_verb,verb_mode)

- Descrizione
- Variabili
- Schema

Funzione print_table()

- Descrizione
- Variabili
- Schema

Funzione get_spaces(lenght_space:int) -> str:

- Descrizione
- Variabili
- Schema

SCHERMATE DI FUNZIONAMENTO (2-3 immagini):

Inizialmente viene chiesto all'utente di inserire le risposte alle domande:

```
Inserisci la forma base di 'Battere': beat
Inserisci il participio passato di 'Awake': awake
Inserisci il simple past di 'Begin': begin
-----
                                RIEPILOGO
-----
| 1 | Battere          | beat          | OK          | 1 punto |
| 2 | Awake             | awake         | NO, Awoken  | 0 punti |
| 3 | Begin             | begin         | NO, Began   | 0 punti |
-----
Hai totalizzato 1 punti su 3
```

Una volta risposto viene visualizzata la tabella con tutte le risposte:

```
-----
                                RIEPILOGO
-----
| 1 | Battere          | beat          | OK          | 1 punto |
| 2 | Awake             | awake         | NO, Awoken  | 0 punti |
| 3 | Begin             | begin         | NO, Began   | 0 punti |
-----
Hai totalizzato 1 punti su 3
```

Il programma finisce stampando il punteggio totale.

CODICE:

Tutto il codice del progetto:

```
#-----  
# Si vuole realizzare un programma che ci aiuti a studiare l'inglese, in particolare i verbi irregolari.  
# Vengono proposti all'utente 10 verbi.  
# La modalità di verifica può cambiare di volta in volta  
# infatti può venir chiesto all'utente di scrivere partendo dalla forma base:  
# - il "Simple past"  
# - il "Participio passato"  
# - la traduzione in italiano  
# - oppure di scrivere la forma base data la traduzione in italiano.  
# Una volta risposto a 10 verbi si visualizza il risultato e la relativa correzione.  
#-----  
  
import random  
  
# Per debug per attivare commenti/stampe extra  
verbose_mode_on: bool = False  
  
def load_verbs(verbs: str) -> list[str]:  
    """  
    Questa funzione ci serve per spezzare la lunga stringa  
    di verbi iniziale in una più maneggevole lista.  
    https://www.w3schools.com/python/ref\_string\_split.asp  
    """  
    my_list = verbs.split("|")  
    #if(verbose_mode_on):  
    #    print("We have " + str(len(my_list)) + " irregular verb")  
    return my_list  
  
  
def get_random_indexs(n: int , n_max: int) -> list[int]:  
    """  
    Questa funzione ci permette di estrarre i (n) numeri random  
    Questi numeri possono andare da 0 a n_max  
    Se elemento è presente in una lista  
    https://www.geeksforgeeks.org/check-if-element-exists-in-list-in-python/  
    """
```

```

my_list: list[int] = []

# Controllo che i numeri siano congrui
if(n > n_max):
    msg = "n is greater then n_max. n={0} n_max={1}".format(n,n_max)
    raise Exception(msg)

# Aggiunto tanti elementi alla mia lista
# quanti sono i numeri richiesti
while(len(my_list)< n):
    t_random: int = random.randrange(n_max)
    is_found: bool = (t_random in my_list)
    if(is_found == False):
        my_list.append(t_random)
return my_list

def get_spaces(lenght_space:int) -> str:
    """
    Questa funzione ci permette restituisce una stringa composta solo da spazi
    lunga quanto il parametro
    """
    result: str = ""
    for line in range(lenght_space):
        result = result + " "
    return result

def retrive_verb(input: str, index: int) -> list[str]:
    """
    Questa funzione ci permette di recuperare dalla stringa del verbo
    sigolo tempo verbale che si desidera
    https://www.w3schools.com/python/ref\_string\_split.asp
    """
    my_list = input.split(",")
    # Controllo che i numeri siano congrui
    if(index > len(input)):
        msg = "index is greater then len(input). index={0} len(input)={1}".format(index,len(input))
        raise Exception(msg)
    return my_list[index]

```



```

# MAIN -----

# Stringa che contiene tutte le informazioni sui verbi

irregular_verbs_string: str =
"Abide, Abode, Abode, Sopportare | Arise, Arose, Arisen, Sorgere | Awake, Awoke, Awoken, Svegliarsi | Be, Was -
Were, Been, Essere | Bear, Bore, Born, Sopportar | Beat, Beat, Beaten, Battere | Become, Became, Become, Diventare | Befall, B
efel, Befallen, Accadere | Beget, Bego, Begotten, Procreare | Begin, Began, Begun, Cominciare | Behold, Behel, Beheld, Vede
re | Bend, Bent, Bent, Curvare"

# Messaggio che viene visualizzato all'utente in base al modo

# Bisogna però settare il parametro che è:

question_to_print: list = []

question_to_print.append("Inserisci la forma base di '{0}': ")      # -> TRADUZIONE
question_to_print.append("Inserisci il simple past di '{0}': ")    # -> FORMA BASE
question_to_print.append("Inserisci il participio passato di '{0}': ") # -> FORMA BASE
question_to_print.append("Inserisci la traduzione di '{0}': ")      # -> FORMA BASE

# lista che contiene tutti i verbi, uno per riga
all_verbs: list[str] = load_verbs(irregular_verbs_string)

# Numero di domande da chiedere all'utente
n_questions: int = 3

# Seleziono gli indici dei verbi da prendere
selected_verbs_index: list[int] = get_random_indexes(n_questions, len(all_verbs)-1)

# Mi serve per contare le domande fatte all'utente
count: int = 0

# Mi indica il tipo di domanda da fare

# 0 > Forma base
# 1 > Simple past
# 2 > Participio passato
# 3 > Traduzione in italiano
verb_mode: int = 0

# Conto i punti fatti
score: int = 0

# Lista dove salvo il risultato della prova
list_result: list[str] = []

# Utili per la stampa del risultato
table_length_message_to_print_parameter: int = 0
table_length_user_answare: int = 0
table_length_answare: int = 0

# numeri dispari sono i dati
# numeri pari sono gli spazi
message_for_result = " | {0} | {1}{2} | {3}{4} | {5}{6} | {7} | "

```

```

while(count < n_questions):

    # Recupero l'indice
    current_random_index = selected_verbs_index[count]

    # Recupero il verbo da chiedere all'utente
    current_verb : str = all_verbs[current_random_index]

    # Estraggo in modo casuale il tipo di esercizio
    verb_mode = random.randrange(0,3)

    # Prendo la parola da chiedere all'utente
    answere: str = retrieve_verb(current_verb,verb_mode)

    # Mi serve per poter mostrare all'utente il messaggio
    message_to_print_parameter: str = ""

    # Mi è uscita forma base. Devo dare all'utente la traduzione in italiano
    if(verb_mode == 0):
        message_to_print_parameter = retrieve_verb(current_verb,3)
    else:
        message_to_print_parameter = retrieve_verb(current_verb,0)
    msg_to_disply :str= (question_to_print[verb_mode]).format(message_to_print_parameter)
    user_answare = input(msg_to_disply).strip()

    # Segnalo la correzione all'utente
    correction: str = ""
    point = "0 punti"

    if(user_answare.lower() == answare.lower()):
        score = score + 1
        correction = "OK"
        point = "1 punto"
    else:
        correction = "NO," + answare

    # Mi aggiornno i contatori per gli spazi nella tabella
    table_length_message_to_print_parameter = 25 - len(message_to_print_parameter)
    table_length_answare = 25 - len(user_answare)
    table_length_user_answare = 30- len(correction)

    # Aggiungo una riga per ogni verbo
    list_result.append(
        message_for_result.format(
            (count+1),
            message_to_print_parameter,
            get_spaces(table_length_message_to_print_parameter),

```

```

        user_answare,
        get_spaces(table_length_answare),
        correction,
        get_spaces(table_length_user_answare),
        point))

    count = count + 1

# Stampo il punteggio finale e facendo il ciclo sulla lista il risultato per ogni verbo
# Mi calcolo la grandezza barra. Grande come la prima riga
header: str = ""
for i in range(len(list_result[0])):
    header = header + "-"

# Stampo l'header della tabella
print(header)

# Mi calcolo la distanza per mettere la scritta a metà schermo
t = int(((len(list_result[0]) - len("RIEPILOGO")))/2)
print(get_spaces(t)+ "RIEPILOGO")
print(header)

# Stampo il risultato per ogni verbo
for line in list_result:
    print(line)

# stampo fine tabella
print(header)
print(header)

# Stampo il punteggio
print("")
print("Hai totalizzato {0} punti su {1}".format(score, n_questions))
print("")

```