

## CASI DI TEST MODULO - GESTIRE\_CLASSIFICA

Funzione aggiornare\_classifica:

Caso di test 1:

Input:

vincitore = "Jasmine"

giocatore = Record("Jasmine", 3)

Risultato atteso:

Creazione del file classifica, Jasmine prima classificata con 3 tiri.

Caso di test 2:

Input:

vincitore = "Max",

giocatore = Record("Max", 5)

Classifica iniziale:

["Jasmine" (3), "Max" (5), "Alex" (9)]

Risultato atteso:

La funzione aggiorna il numero di tiri di "Max" a 5 e aggiorna la classifica nel file di testo.

Caso di test 3:

Input:

vincitore = "Bull",

giocatore = Record("Bull", 7)

Classifica iniziale:

["Jasmine" (3), "Max" (5), "Bull" (7), "Alex" (9)]

Risultato atteso:

La funzione inserisce "Bull" come nuovo classificato, inserendolo al posto corretto.

Funzione scrivere\_classifica\_vuota :

Caso di test 1:

Input:

```
vincitore = Record("Jasmine ", 3),  
lista_classificati = []
```

Risultato atteso:

Viene creato il file della classifica e inserito il vincitore come primo elemento della classifica.

Caso di test 2:

Input:

```
vincitore = Record("Max", 5),  
lista_classificati = ["Jasmine " (3), " Max " (5), "Alex"  
(9)]
```

Risultato atteso:

Non viene creato un nuovo file di classifica, si riscrive la classifica nel file di testo e viene restituito lista\_classificati con il vincitore.

Caso di test 3:

Input:

```
vincitore = Record("Bull", 7),  
lista_classificati = []
```

Risultato atteso:

La funzione non scrive nel file di classifica, restituisce lista\_classificati vuota e si verifica un errore.

#### Caso di test 4:

Input:

```
vincitore = Record("Fabio", 0)
```

```
lista_classificati = []
```

Risultato atteso:

Viene creato il file della classifica, inserito il vincitore con zero tiri come primo elemento della classifica.

#### Caso di test 5:

Input:

```
vincitore = Record("Dusan",19)
```

```
lista_classificati = ["Jasmine " (3), " Max " (5), "Alex" (9)]
```

Risultato atteso:

La funzione non crea un nuovo file di classifica, restituisce lista\_classificati con il vincitore.

#### Funzione inserire\_classifica:

##### Caso di test 1:

Input:

```
vincitore = Record("Alex", 2)
```

```
lista_classificati = ["Fabio" (5) "Niga" (6), "Dave" (8)]
```

Risultato atteso:

Inserisce il vincitore come primo classificato nella lista nella posizione corretta.

##### Caso di test 2:

Input:

```
vincitore = Record("Luca", 9)
```

```
lista_classificati = ["Fabio" (5) "Niga" (6), "Dave" (8),  
"Steve" (9)]
```

Risultato atteso:

La funzione inserisce il vincitore nella posizione immediatamente successiva al classificato con lo stesso numero di tiri.

### Caso di test 3:

Input:

```
vincitore = Record("Franco", 12)
```

```
lista_classificati = ["Fabio" (5) "Niga" (6), "Dave" (8),  
"Dave" (9)]
```

Risultato atteso:

La funzione non modifica la lista\_classificati perché il vincitore ha un numero di tiri più alto rispetto agli altri giocatori classificati.

### Caso di test 4:

Input:

```
vincitore = Record("Jasmine", 3)
```

```
lista_classificati = []
```

Risultato atteso:

La funzione inserisce il vincitore come unico classificato nella lista vuota.

Funzione aggiornare\_classificato:

### Caso di test 1:

Input:

```
pos_classificato = 2,  
vincitore = Record("Nico", 5)  
lista_classificati = ["Jasmine" (6), "Giuseppe" (7),  
"Alice" (8), "Davide" (10)]
```

Risultato atteso:

La funzione confronta il numero di tiri del secondo classificato con il numero di tiri del vincitore. Il vincitore è inserito nella lista\_classificati nella posizione corretta perché ha un numero inferiore di numeri dei lanci.

#### Caso di test 2:

Input:

```
pos_classificato = 2,  
vincitore = Record("Emiliano", 7)`,  
lista_classificati = ["Jasmine" (6), "Giuseppe" (7),  
"Alice" (8), "Davide" (10)]
```

Risultato atteso:

La funzione confronta il numero di tiri del secondo classificato con il numero di tiri del vincitore. Siccome il numero dei tiri è uguale non viene aggiornata la lista\_classifica.

#### Caso di test 3:

Input:

```
pos_classificato = 4,  
vincitore = Record("Arianna", 9)`,  
lista_classificati = ["Jasmine" (6), "Giuseppe" (7),  
"Alice" (8), "Davide" (10)]
```

Risultato atteso:

La funzione confronta il numero di tiri del quarto classificato con il numero di tiri del vincitore. Siccome il numero dei tiri è inferiore, il vincitore viene inserito nella lista\_classifica nella posizione corretta.

#### Caso di test 4:

Input:

```
pos_classificato = 0,  
vincitore = Record("Alice", 3)`,  
lista_classificati = ["Jasmine" (6), "Giuseppe" (7),  
"Alice" (8), "Davide" (10)]
```

Risultato atteso:

La funzione confronta il numero di tiri del classificato in posizione 0 con il numero di tiri del vincitore. Siccome è il vincitore stesso ad essere nella posizione 0, non viene aggiornata la lista\_classifica.

#### Caso di test 5:

Input:

```
pos_classificato = NO_RICERCA,  
vincitore = Record("Jasmine", 5)  
lista_classificati = []`
```

Risultato atteso:

La funzione non effettua alcuna modifica alla lista poiché non è presente alcun classificato. La lista\_classificati non viene modificata.

#### Funzione convertire\_giocatore:

#### Caso di test 1:

Input:

```
giocatore = Record("Alessia", 3) `
```

Risultato atteso:

La funzione legge il nome del giocatore e lo inserisce nel campo "nome\_classificato" del record classificato; legge il numero di tiri del giocatore e lo inserisce nel campo "lanci " nel record classificato.

#### Caso di test 2:

Input:

```
giocatore = Record("", 5)
```

Risultato atteso:

La funzione non legge il nome del giocatore poiché è vuoto. Inserisce il carattere di fine stringa nel campo "nome\_classificato" del nuovo classificato; legge il numero di tiri del giocatore e lo inserisce nel campo "lanci" del record classificato.

#### Caso di test 3:

Input:

```
giocatore = Record("A", 8) `
```

Risultato atteso:

La funzione legge il nome del giocatore "A" e lo inserisce nel campo "nome\_classificato" del record classificato; legge il numero di tiri del giocatore e lo inserisce nel campo "lanci " del record classificato.

#### Funzione ricercare\_classificati:

##### Caso di test 1:

Input:

```
file_classifica: ["Jasmine", 4], ["Oshimen", 7], ["Bob", 3]
```

Risultato atteso:

La funzione legge il file\_classifica e li inserisce nell'array lista\_classificati nella posizione corretta.

#### Caso di test 2:

Input:

```
file_classifica: vuoto
```

Risultato atteso:

La funzione non legge nulla dal file perché è vuoto.

Restituisce la lista\_classificati vuota.

#### Caso di test 3:

Input:

```
file_classifica: ["Simba", 5]
```

Risultato atteso:

La funzione legge il file\_classifica e lo inserisce nella lista\_classificati. Restituisce lista\_classificati con il l'unico classificato.

#### Funzione confrontare\_lista:

#### Caso di test 1:

Input:

```
lista_classifica: ["Shiva", 4], ["Sfera", 6], ["Charlie", 3]
```

vincitore: "Sfera"

Risultato atteso:



La funzione confronta il nome del vincitore con i nomi presenti nella lista dei classificati. Trova una corrispondenza con il nome nella posizione 2 della lista dei classificati. Restituisce `pos\_corrispondenza` con il valore 2.

#### Caso di test 2:

Input:

```
lista_classifica: ["Shiva", 4], ["Sfera", 6], ["Charlie", 3]
```

vincitore: "Simba"

Risultato atteso:

La funzione confronta il nome del vincitore con i nomi presenti nella lista\_classificati. Non trova corrispondenze e restituisce pos\_corrispondenza con il valore `NO\_RICERCA`.

#### Caso di test 3:

Input:

```
lista_classifica: vuota
```

vincitore: "Alice"

Risultato atteso:

Siccome la lista\_classificati è vuota la funzione non trova corrispondenze. Restituisce pos\_corrispondenza con il valore `NO\_RICERCA`.

#### Caso di test 4:

Input:

```
lista_classifica: ["Shiva", 4], ["Sfera", 6], ["Charlie", 3], ["Leo", 8],
```

vincitore: "Leo"

Risultato atteso:

La funzione confronta il nome del vincitore con i nomi presenti nella lista dei classificati. Trova una

corrispondenza con il nome nella posizione 4 della lista dei classificati. Restituisce `pos\_corrispondenza` con il valore 4.

#### Caso di test 5:

Input:

lista\_classifica: vuota

vincitore: ("")

Risultato atteso:

Siccome la lista\_classificati è vuota e anche il vincitore è vuoto la funzione restituisce pos\_corrispondenza con il valore `NO\_RICERCA`.

#### Funzione confrontare\_numero\_tiri:

#### Caso di test 1:

Input:

lista\_classifica: vuota

vincitore` con numero di tiri 7

Risultato atteso:

Siccome la lista\_classifica è vuota, la funzione restituisce `pos\_corrispondenza` con il valore `NO\_RICERCA`.

#### Caso di test 2:

Input:

lista\_classifica:["Alice", 5]

vincitore` con numero di tiri 5

Risultato atteso:

La funzione confronta il numero di tiri del vincitore con il numero di tiri dei classificati presenti nella lista. Trova una corrispondenza e restituisce ``pos_corrispondenza`` con il valore 1.

#### Caso di test 3:

Input:

`lista_classifica: ["Alice", 3]`

`vincitore`` con numero di tiri 5

Risultato atteso:

La funzione confronta il numero di tiri del vincitore con il numero di tiri dei classificati presenti nella lista. Siccome non trova corrispondenze, restituisce ``pos_corrispondenza`` con il valore ``NO_RICERCA``.

#### Caso di test 4:

Input:

`lista_classifica: ["Ghali", 5], ["TaxiB", 7], ["Tony", 4], ["Davide", 6]`

`vincitore`` con numero di tiri 5

Risultato atteso:

La funzione confronta il numero di tiri del vincitore con i numeri di tiri presenti nella lista dei classificati. Trova una corrispondenza in cui il numero di tiri del classificato è maggiore del vincitore nella posizione 2 della lista dei classificati. Restituisce ``pos_corrispondenza`` con il valore 2.

Funzione `scrivere_classifica:`

#### Caso di test 1:

Input:

file\_classifica: vuoto

lista\_classifica: vuota

Risultato atteso:

La funzione non scrive sul file della classifica e restituisce file\_classifica senza modifiche.

#### Caso di test 2:

Input:

file\_classifica` contenente dei record di classificati

lista\_classifica: ["Simba", 2]

Risultato atteso:

La funzione scrive la lista\_classifica sul file della classifica e restituisce file\_classifica aggiornato.

#### Caso di test 3:

Input:

file\_classifica contenente dei record di classificati

lista\_classifica = ["Paky", 4], ["Shiva", 5], ["Charlie", 6]

Risultato atteso:

La funzione scrive la lista\_classifica sul file della classifica in sequenza e restituisce `file\_classifica` aggiornato.

#### Caso di test 4:

Input:

file\_classifica: vuoto

lista\_classifica: ["Alice", 5]

Risultato atteso: La funzione scrive il record del classificato ["Alice", 5] sul file della classifica e restituisce `file\_classifica` con il nuovo record.

### Caso di test 5:

Input:

file\_classifica` contenente dei record di classificati

lista\_classifica: vuota

Risultato atteso:

La funzione cancella tutti i record presenti nel file della classifica, lasciando il file vuoto, e restituisce file\_classifica senza modifiche.

### Funzione inserire\_elemento:

#### Caso di test 1:

Input:

lista\_classifica: vuota

classificato: ["Shiva", 3]

pos\_inserimento: 0

Risultato atteso:

lista\_classificati diventa ["Shiva", 3] come unico elemento.

#### Caso di test 2:

Input:

lista\_classifica: ["Shiva", 3], ["Evelyn", 4], ["Paky", 5]

classificato: ["Davide", 2]

pos\_inserimento: 1

Risultato atteso:

lista\_classificati ["Davide", 2], ["Shiva", 3], ["Evelyn", 4]

### Caso di test 3:

Input:

lista\_classifica: ["Keys", 4], [FINE\_LISTA]

classificato: ["Max", 3]

pos\_inserimento:1

Risultato atteso:

lista\_classificati ["Keys", 4], ["Max", 3], [FINE\_LISTA].

### Caso di test 4:

Input:

lista\_classificati: ["Shiva", 4], ["Max", 3],  
[FINE\_LISTA]

classificato: ["Giulia", 6]

pos\_inserimento: 0

Risultato atteso: `lista\_classificati` diventa ["Giulia",  
6], ["Shiva", 4], ["Max", 3].

Funzione scrivere\_classifica\_testo:

### Caso di test 1:

Input:

lista\_classificati: vuota

Risultato atteso:

flag = 1 e il file di testo rimane vuoto.

### Caso di test 2:

Input:

```
lista_classificati: ["Shiva", 7], ["Rondo", 4], ["Simba", 5], [FINE_LISTA]
```

Risultato atteso:

flag = 1 e il file di testo contiene il seguente contenuto:

```
!!! CLASSIFICA VINCITORI !!!
```

<i>Pos</i>	<i>Nome</i>	<i>Lanci</i>
<i>1</i>	<i>Shiva</i>	<i>7</i>
<i>2</i>	<i>Rondo</i>	<i>4</i>
<i>3</i>	<i>Simba</i>	<i>5</i>

Caso di test 3:

Input:

```
lista_classificati: ["Shiva", 7], ["Rondo", 4], [FINE_LISTA], [FINE_LISTA]
```

Risultato atteso:

flag = 1 e il file di testo contiene il seguente contenuto:

```
!!! CLASSIFICA MIGLIORI OCHE !!!
```

<i>Pos</i>	<i>Nome</i>	<i>N.Tiri</i>
<i>1</i>	<i>Shiva</i>	<i>7</i>
<i>2</i>	<i>Rondo</i>	<i>4</i>

Caso di test 4:

Input:

```
lista_classificati: ["Shiva", 7], ["Rondo", 4], ["Simba", 5], [FINE_LISTA]
```

```
file_classifica_testo: restituisce `ERRORE_FILE`
```

Risultato atteso:

flag = 0 e il file di testo rimane vuoto.

Caso di test 5:

Input:

lista\_classificati: vuota

file\_classifica\_testo: restituisce 0

Risultato atteso:

flag = 0 e il file di testo rimane vuoto.

Funzione aggiornamento\_classifica:

Caso di test 1:

Input:

vincitore con numero di tiri = 3

Risultato atteso:

flag = 0

Caso di test 2:

Input:

vincitore con numero di tiri = PARTITA\_INTERROTTA

Risultato atteso:

flag != 0

Caso di test 3:

Input:

vincitore con numero di tiri = ERRORE\_FILE

Risultato atteso:

flag = ERRORE\_FILE.

Caso di test 4:

Input:

vincitore con numero di tiri = 6



```
flag: ERRORE_FILE
```

Risultato atteso:

```
flag = ERRORE_FILE.
```

Caso di test 5:

Input:

```
vincitore con numero di tiri = 5
```

```
flag != ERRORE_FILE
```

Risultato atteso:

```
flag = 0.
```