

MODULO - GESTIRE_CLASSIFICA

Moduli Inclusi:

GESTIRE_FILE
GESTIRE_PARTITE
GESTIRE_CLASSIFICA

Costanti:

NO_RICERCA = -1
FINE_LISTA = -1
MAX_CLASSIFICATI = 10

Funzione scrivere_nome_classifica

INPUT:

classifica, classificato di cui inserire un carattere del nome nel campo nome_classifica di classifica, Record classifica

pos, posizione in cui inserire il carattere del nome, intero
>= 0 <= 4

c, carattere da inserire in posizione pos del campo nome_classifica di classifica, carattere

OUTPUT:

classificaa, classificato di cui è stato aggiornato il campo nome_classifica di classifica, Record classifica

INZIO

INPUT

classifica, classificato di cui leggere un carattere del nome nel campo nome_classifica di classifica, Record classifica

pos, posizione di cui leggere il carattere del nome, Intero,
 $0 \leq \text{pos} \leq 3$

OUTPUT

c, carattere letto in posizione pos del campo nome_classificato del record classifica, carattere

INIZIO

c = elemento in posizione pos del campo nome_classifica di classifica

FINE

Funzione scrivere_lanci

INPUT:

classifica, classificato di cui inserire il numero di tiri effettuati, Record classifica

lanci, numero di tiri effettuati, intero ≥ 0

OUTPUT:

classifica, classificato di cui è stato aggiornato il campo lanci, Record classifica

INIZIO

elemento del campo lanci di classifica = lanci

FINE

Funzione leggere_lanci

INPUT:

classifica, classificato di cui leggere il numero di tiri effettuati, Record classifica

OUTPUT:

num_tiri, numero di lanci effettuati, intero ≥ 0

INIZIO

num_tiri = elemento del campo lanci di classifica

FINE

Funzione aggiornare_classifica

INPUT:

vincitore, vincitore dell'ultima partita, Record classifica

OUTPUT:

flag, segnale di corretta apertura dei file, Intero

LAVORO:

file_classifica, file classifica aggiornato, FILE binario

lista_classificati, array di Record classifica contenente tutti i classificati, Array di Record classifica

vincitore_convertito, conversione del Record giocatore vincitore nella corrispondente forma semplificata di classifica, Record classifica

pos, posizione della lista_classificati, intero ≥ 0

verifica, numero di tiri del primo classificato della
lista, intero > 0

elemento_0, primo classificato nella lista dei
classificati, Record classifica

INIZIO

flag = 1

file_classifica = ricercare_slots(CLASSIFICA_BINARIA,
LETTURA_BINARIA)

SE (file_classifica = 0)

file_classifica = ricercare_slots(CLASSIFICA_BINARIA,
SCRITTURA_BINARIA)

file_classifica = ricercare_slots(CLASSIFICA_BINARIA,
LETTURA_BINARIA)

SE (file_classifica = 0)

flag = ERRORE_FILE

fine SE

fine SE

SE (flag != ERRORE_FILE)

lista_classificati =
ricercare_classificati(file_classifica, lista_classifica
ti)

vincitore_convertito =
convertire_giocatore(vincitore, vincitore_convertito)

elemento_0 = elemento in posizione 0 di lista_classificati

verifica = leggere_lanci_classifica(elemento_0)

SE (verifica = FINE_LISTA)

```

lista_classificati =
scrivere_classifica_vuota(vincitore_convertito,
lista_classificati)

ALTRIMENTI

pos = confrontare_lista(lista_classificati,
vincitore_convertito)

SE (pos = NO_RICERCA)

lista_classificati =
inserire_classifica(vincitore_convertito,
lista_classificati)

ALTRIMENTI

lista_classificati = aggiornare_classificato(pos,
vincitore_convertito, lista_classificati)

fine SE

fine SE

SE (flag != ERRORE_FILE)

flag = scrivere_classifica_testo(lista_classificati)

fine SE

fine SE

FINE

```

Funzione scrivere_classifica_vuota

INPUT:

vincitore, giocatore vincitore, Record classifica

lista_classificati, lista di classificati, Array di Record
classifica

OUTPUT:

lista_classificati, lista di classificati, Array di Record
classifica

LAVORO:

file_classifica, file contenente la classifica, FILE
binario

INIZIO

file_classifica = ricercare_slots(CLASSIFICA_BINARIA,
SCRITTURA_BINARIA)

lista_classificati = inserire_elemento
(lista_classificati, vincitore, 0)

scrivere_classifica(file_classifica, lista_classificati)

FINE

Funzione inserire_classifica

INPUT:

vincitore, vincitore dell'ultima partita, Record
classifica

lista_classificati, lista di classificati, Array di Record
classifica

OUTPUT:

lista_classificati, lista di classificati aggiornata,
Array di Record classifica

LAVORO:

pos, posizione in cui si trova il classificato con un numero
di tiri maggiore dell'attuale vincitore, intero ≥ 0

file_classifica, file contenente la classifica, FILE
binario

INIZIO

```
pos = confrontare_numero_tiri(lista_classificati,
vincitore)

SE (pos != NO_RICERCA)

lista_classificati = inserire_elemento
(lista_classificati, vincitore, pos)

file_classifica = ricercare_slots(CLASSIFICA_BINARIA,
SCRITTURA_BINARIA)

scrivere_classifica(file_classifica, lista_classificati)

FINE
```

Funzione aggiornare_classificato

INPUT:

```
pos_classificato, posizione del classificato, intero, >=
0

vincitore, vincitore da inserire nella lista dei
classificati, Record classifica

lista_classificati, lista di classificati, Array di Record
classifica
```

OUTPUT:

```
lista_classificati, lista di classificati, Array di Record
classifica
```

LAVORO:

```
corrispondente, classificato con lo stesso nome del
vincitore, Record classifica
```

num_tiri_classifica, numero di tiri del classificato
corrente, intero > 0

num_tiri_vincitore, numero di tiri del vincitore, intero
> 0

file_classifica, file contenente la classifica, FILE
binario

INIZIO

corrispondente = elemento in posizione pos_classificato di
lista_classificati

num_tiri_classifica =
leggere_lanci_classificato(corrispondente)

num_tiri_vincitore = leggere_lanci_vincitore(vincitore)
SE (num_tiri_classifica > num_tiri_vincitore)

pos_classificato =
confrontare_numero_tiri(lista_classificati, vincitore)

SE (pos_classificato != NO_RICERCA)

inserire_elemento (lista_classificati, vincitore,
pos_classificato)

fine SE

file_classifica =
ricercare_slot(RIGA_CLASSIFICA_BINARIA,
SCRITTURA_BINARIA)

scrivere_classifica(file_classifica, lista_classificati)

fine SE

FINE

Funzione convertire_giocatore

INPUT:

giocatore, giocatore da convertire a classificato, Record
giocatore

OUTPUT:

classificato, giocatore convertito, Record classifica

LAVORO:

i, indice del nome del giocatore da convertire, intero \geq
0

c, carattere letto all'interno del campo in posizione i,
Carattere

lanci, numero di tiri svolti dal giocatore vincitore,
Intero > 0

INIZIO

i = 1

MENTRE (i < LUNGH_NOME -1)

c = leggere_nome_giocatore(giocatore, i)

classificato = scrivere_nome_classificato(classificato,
i, c)

i = i + 1

fine MENTRE

classificato = scrivere_nome_classifica(classificato, i,
FINE_STRINGA)

lanci = leggere_lanci(giocatore)

classificato = scrivere_lanci_classifica(classificato,
lanci)

FINE

Funzione ricercare_classificati

INPUT:

file_classifica, file contenente i record dei
classificati, FILE binario

lista_classificati, lista di classificati, Array di Record
classifica

OUTPUT:

lista_classificati, array di classificati inseriti nella
classifica, Array di Record classifica

LAVORO:

i, indice della array lista_classifica, Intero ≥ 0

verifica, numero di elementi letti dal file, Intero ≥ 0

INIZIO

i = 1

verifica = calcolare_lunghezza_file(file_classifica)

MENTRE (verifica != 0)

i = i + 1

elemento in posizione i di lista_classificati =
leggere_file(file_classifica)

verifica = calcolare_lunghezza_file(file_classifica)

fine MENTRE

lista_classificati = scrivere_lanci_classifica(elemento
in posizione i di lista_classificati, FINE_LISTA)

FINE

Funzione confrontare_lista

INPUT:

lista_classifica, array di classificati inseriti nella classifica, Array di Record classifica

vincitore, vincitore dell'ultima partita, Record classifica

OUTPUT:

pos_corrispondenza, posizione della lista dei classificati in cui vi è la corrispondenza con il nome del vincitore, intero

LAVORO:

i, indice del nome del vincitore, Intero ≥ 0

nome_vincitore, stringa contenente il nome del giocatore vincitore, Array di Caratteri terminato con il carattere di FINE_STRINGA

pos, posizione all'interno della lista dei classificati, Intero ≥ 0

verifica, variabile che indica se è stata trovata una corrispondenza con il nome del vincitore all'interno della classifica, Intero =0 o =1

classificato, giocatore in classifica che viene letto, Record classifica

nome_classificato, stringa contenente il nome del classificato, Array di Caratteri terminato con il carattere di FINE_STRINGA

lanci_classifica, numero di tiri effettuati dal classificato, Intero

INIZIO

pos_corrispondeza = NO_RICERCA

i = 1

MENTRE (i < LUNGH_NOME)

elemento in posizione i di nome_vincitore =
leggere_nome_classifica(vincitore, i)

i = i + 1

fine MENTRE

pos = 1

verifica = 1

ESEGUI

classificato = elemento in posizione pos di
lista_classifica

i = 1

MENTRE (i < LUNGH_NOME)

elemento in posizione i di nome_classificato =
leggere_nome_classifica(elemento in posizione pos di
lista_classifica, i)

i = i + 1

fine MENTRE

verifica = confrontare_stringhe(nome_vincitore,
nome_classificato)

SE (verifica = 0)

pos_corrispondenza = pos

fine SE

lanci_classifica =
leggere_lanci_classifica(classificato)

pos = pos + 1

```
FINCHÈ((lanci_classifica != FINE_LISTA) && (verifica != 0))
```

FINE

Funzione confrontare_numero_tiri:

INPUT:

lista_classifica, array di classificati inseriti nella classifica, Array di Record classifica

vincitore, vincitore dell'ultima partita, Record classifica

OUTPUT:

pos_corrispondenza, posizione della lista dei classificati in cui vi è la corrispondenza con il nome del vincitore, intero

LAVORO:

lanci_vincitore, numero di tiri effettuati dal giocatore vincitore, Intero > 0

pos, posizione all'interno della lista dei classificati, Intero >= 0

verifica, variabile che indica se è stata trovata una corrispondenza con il numero, Intero

classificato, giocatore in classifica letto, Record classifica

lanci_classifica, numero di tiri effettuati dal classificato attualmente in confronto, Intero

INIZIO

```
lanci_vincitore = leggere_lanci_classifica(vincitore)
pos = 1
verifica = 0
ESEGUI
classificato = elemento in posizione pos di
lista_classifica
lanci_classifica =
leggere_lanci_classifica(classificato)
SE (lanci_classifica > lanci_vincitore) OR
(lanci_classifica = FINE_LISTA)
verifica = 1
pos_corrispondenza = pos
fine SE
pos = pos + 1
FINCHÈ ((lanci_classifica != FINE_LISTA) AND (verifica !=
1)) AND (pos < MAX_CLASSIFICATI -1)
SE (verifica = 0)
pos_corrispondenza = NO_RICERCA
fine SE
FINE
```

Funzione scrivere_classifica:

INPUT:

file_classifica, file contenente i record dei
classificati, FILE binario

lista_classifica, array di classificati inseriti nella classifica, Array di Record classifica

OUTPUT:

file_classifica, file contenente i record dei classificati, FILE binario

LAVORO:

pos, posizione all'interno della lista dei classificati, Intero ≥ 0

verifica, valore corrispondente al numero di tiri del classificato attualmente letto, Intero

INIZIO

pos = 1

verifica = leggere_lanci_classifica(elemento in posizione pos di lista_classifica)

MENTRE (verifica != FINE_LISTA)

file_classifica = scrivere_file(file_classifica, elemento in posizione pos di lista_classifica)

pos = pos + 1

verifica = leggere_lanci_classifica(elemento in posizione pos di lista_classifica)

fine MENTRE

file_classifica = scrivere_file(file_classifica, elemento in posizione pos di lista_classifica)

FINE

Funzione inserire_elemento

INPUT:

lista_classifica, array di classificati inseriti nella classifica, Array di Record classifica

classificato, classificato da inserire nell'array lista classifica, Record classifica

pos_inserimento, posizione in cui inserire l'elemento, Intero

OUTPUT:

lista_classifica, array di classificati inseriti nella classifica, Array di Record classifica

LAVORO:

pos_lista, indice all'interno della lista dei classificati, Intero, ≥ 0

verifica, valore corrispondente al numero di tiri del classificato attualmente letto, Intero

INIZIO

pos_lista = 1

verifica = leggere_lanci_classifica(elemento in posizione pos_lista di lista_classifica)

MENTRE ((verifica \neq FINE_LISTA) AND (pos_lista < MAX_CLASSIFICATI))

pos_lista = pos_lista + 1

verifica = leggere_lanci_classifica(elemento in posizione pos_lista di lista_classifica)

fine MENTRE

SE (pos_lista = 0)


```

elemento in posizione pos_lista + 1 di lista_classifica =
elemento in posizione pos_lista di lista_classifica

ALTRIMENTI

SE (pos_lista = MAX_CLASSIFICATI -1)
pos_lista = pos_lista - 1
fine SE

MENTRE (pos_lista > pos_inserimento)
elemento in posizione pos_lista di lista_classifica =
elemento in posizione pos_lista - 1 di lista_classifica
pos_lista = pos_lista - 1
fine MENTRE

fine ALTRIMENTI

elemento in posizione pos_inserimento di lista_classifica
= classificato
FINE

```

Funzione scrivere_classifica_testo:

INPUT:

lista_classificati, array di classificati inseriti nella classifica, Array di Record classifica

OUTPUT:

flag, indica l'esito dell'operazione di scrittura, intero

LAVORO:

file_classifica_testo, file classifica aggiornato, FILE di testo

pos, posizione all'interno della lista dei classificati, Intero >= 0

classificato, classificato in posizione pos nella lista,
Record classifica

num_tiri, numero di tiri del classificato, Intero >= 0

i, indice nell'array del nome del classificato, Intero >= 0

nome_classificato, array contenente il nome del
classificato, Array di Caratteri che termina con il
carattere di FINE_STRINGA

INIZIO

```
TITOLO_CLASSIFICA[] =  
"+-----+\n|          CLASSIFICA
```

```
VINCITORI
```

```
|\n+-----+"
```

```
LAYOUT[] = "+-----+-----+-----+"
```

```
LARGHEZZA_NOME = 15
```

```
flag = 1
```

```
file_classifica_testo = ricercare_il_menu(CLASSIFICA,  
SCRITTURA_TESTO)
```

```
SE (file_classifica_testo = 0)
```

```
flag = ERRORE_FILE
```

```
ALTRIMENTI
```

```
stampare_a_file(file_classifica_testo,  
TITOLO_CLASSIFICA)
```

```
stampare_a_file(file_classifica_testo, LAYOUT)
```

```
pos = 0
```

```
num_tiri = 0
```

```
ESEGUI
```

```

classificato = elemento in posizione pos di
lista_classificati
num_tiri = leggere_lanci_classifica(classificato)
SE (num_tiri != FINE_LISTA)
i = 1
MENTRE (i < LUNGH_NOME)
elemento in posizione i di nome_classificato =
leggere_nome_classifica(elemento in posizione pos di
lista_classifica, i)
i = i + 1
fine MENTRE
stampare_a_file(file_classifica_testo,pos+1,LARGHEZZA_N
OME,nome_classificato,num_tiri)
fine SE
pos = pos + 1
FINCHÈ (num_tiri != FINE_LISTA)
stampare_a_file(file_classifica_testo,pos+1,LAYOUT)
fine ALTRIMENTI
FINE

```

Funzione aggiornamento_classifica:

INPUT:

vincitore, giocatore che vince la partita in corso, Record
giocatore

OUTPUT:

flag, flag che indica l'esito dell'aggiornamento, intero

LAVORO:

validita, numero di tiri del vincitore, Intero

INIZIO

validita = leggere_lanci(vincitore)

SE ((validita != PARTITA_INTERROTTA) AND (validita !=
ERRORE_FILE))

flag = aggiornare_classifica(vincitore)

fine SE

SE (validita = ERRORE_FILE)

flag = ERRORE_FILE

ALTRIMENTI SE (flag != ERRORE_FILE)

flag = 0

fine SE

FINE