

MODULO GESTIRE_FILE

Moduli inclusi:

GESTIRE_FILE

Costanti:

MAX_PERCORSO_FILE = 257

Algoritmi:

Funzione leggere_file_di_testo:

INPUT:

file_testo, file di testo da leggere, file di testo.

num_caratteri, dimensione dell'array stringa_file, intero > 0.

stringa_file, array di caratteri contenente tutti i caratteri del file, array monodimensionale di caratteri terminato con il carattere di FINE_STRINGA.

OUTPUT:

stringa_file, array di caratteri contenente tutti i caratteri del file, array monodimensionale di caratteri terminato con il carattere di FINE_STRINGA.

LAVORO:

i, indice di stringa_file, intero > 0

c, carattere che viene letto e memorizzato nella stringa_file, carattere

INIZIO

```
i = 1
MENTRE (i < num_caratteri)
c = leggere_carattere_file(file_testo)
elemento in posizione i di stringa_file = c
i = i + 1
fine MENTRE
elemento in posizione i di stringa_file = FINE_STRINGA
FINE
```

Funzione scrivere_file_di_testo:

INPUT:

file_testo, file di testo su cui scrivere, file di testo
stringa, stringa da trascrivere sul file di testo, array
monodimensionale di caratteri terminata con il carattere
di FINE_STRINGA

OUTPUT:

file_testo, file modificato con il contenuto della stringa,
file di testo

LAVORO:

i, indice della stringa da stampare, intero > 0

INIZIO

```
i = 1
MENTRE (elemento in posizione i di stringa != FINE_STRINGA)
file_testo = stampare_su_file(file_testo, elemento in
posizione i di stringa)
i = i + 1
```

fine MENTRE

FINE

Funzione `calcolare_lunghezza_file_di_testo`:

INPUT:

file, file di testo da calcolare la lunghezza, file di testo.

OUTPUT:

num_caratteri, numero di caratteri contenuti nel file,
intero ≥ 0

LAVORO:

c, carattere che memorizza il carattere da file, carattere

INIZIO

num_caratteri = 0

c = leggere_carattere_file(file)

MENTRE (c != FINE_FILE)

num_caratteri = num_caratteri + 1

c = leggere_carattere_file(file)

fine MENTRE

FINE

Funzione `stampare_file_di_testo`:

INPUT:

file_testo, file di testo da stampare a video, file di testo

OUTPUT:

//nessuno

LAVORO:

lungh_file, numero di caratteri contenuti nel file incluso quello di FINE_STRINGA, intero > 0

stringa_file, array contenente tutti i caratteri contenuti nel file, array di caratteri terminato con il carattere di FINE_STRINGA

INIZIO

lungh_file= calcolare_lunghezza_file_di_testo(file_testo)

stringa_file=leggere_file_di_testo(file_testo,
lungh_file)

stampare_stringa_da_file(stringa_file)

FINE

Funzione stampare_stringa_da_file:

INPUT:

stringa, stringa di caratteri, array di caratteri terminati con il carattere di FINE_STRINGA

OUTPUT:

//nessuno

LAVORO:

i, indice di scorrimento della stringa, intero > 0

c, carattere letto all'interno della stringa, carattere

INIZIO

i = 1

c = elemento in posizione i di stringa

```
MENTRE (c != FINE_STRINGA)
STAMPARE_A_VIDEO(c)
i = i + 1
c = elemento in posizione i di stringa
fine MENTRE
FINE
```

Funzione ricercare_slots:

INPUT:

slot, numero dello slot della partita scelto dall'utente,
intero $0 \leq \text{MAX_SLOT}$

tipo_apertura, array di caratteri che indica il tipo di
utilizzo che si intende effettuare del file, array di
caratteri

OUTPUT:

file_slot, file contenente il singolo record partita
associato allo slot scelto dall'utente, file binario.

LAVORO:

file_percorsi_salvataggi, file contenente i percorsi di
ciascun file partita, file di testo

percorso, stringa contenente il percorso del file
desiderato, array di caratteri terminato con il carattere
di FINE_STRINGA

INIZIO

```
PERCORSO_BINARI="../../file//lista_file_bin.txt"
```

```
file_percorsi_salvataggi = 0
```

```
file_slot = 0
```

```
file_percorsi_salvataggi=aprire_file(PERCORSO_BINARI,
LETTURA_TESTO)

SE (file_percorsi_salvataggi != 0)

file_percorsi_salvataggi=spostare_a_riga(file_percorsi_salvataggi,slot)

percorso=leggere_riga_di_file(file_percorsi_salvataggi,percorso)

file_slot = aprire_file(percorso, tipo_apertura)

fine SE

FINE
```

Funzione ricercare_il_menu:

INPUT:

riga, riga in cui è contenuto il percorso del menu scelto,
intero >= 0

tipo_apertura, array di caratteri che indica il tipo di
utilizzo che si intende effettuare del file, array di
caratteri

OUTPUT:

file_menu, file di testo che contiene il menu, file di
testo

LAVORO:

file_percorsi_menu, file contenente i percorsi di ciascun
menu, file di testo

percorso, stringa contenente il percorso del file
desiderato, array di caratteri terminato con il carattere
di FINE_STRINGA

INIZIO

```
PERCORSO_MENU="../menu//lista_menu.txt"
file_percorsi_menu = 0
file_menu = 0
file_percorsi_menu=aprire_file(PERCORSO_MENU,LETTURA_TEST
0)
SE (file_percorsi_menu != 0)
file_percorsi_menu = spostare_a_riga(file_percorsi_menu,
riga)
percorso = leggere_riga(file_percorsi_menu,percorso)
file_menu = aprire_file(percorso, tipo_apertura)
fine SE
FINE
```

Funzione spostare_a_riga:

INPUT:

file, file in cui spostarsi, file di testo
riga_attuale, riga a cui spostarsi, intero > 0

OUTPUT:

file, file in cui ci si è mossi alla riga desiderata, file
di testo

LAVORO

riga, riga del file letta, intero > 0
c, carattere letto dal file, carattere

INIZIO

```
riga = 1
SE (riga_attuale != 1)
```

```
c = leggere_da_file(file)
MENTRE ((riga != riga_attuale) AND (c != FINE_FILE))
c = leggere_da_file(file)
MENTRE ((c != LINEA) AND (c != FINE_FILE))
c = leggere_da_file(file)
fine MENTRE
riga = riga + 1
fine MENTRE
fine SE
FINE
```

Funzione leggere_riga_di_file:

INPUT:

file, file di cui leggere la riga, file di testo

riga, stringa contenente il contenuto della riga, array di caratteri terminato con il carattere di FINESTRINGA

OUTPUT:

riga, stringa contenente il contenuto della riga, array di caratteri terminato con il carattere di FINESTRINGA

LAVORO:

c, carattere letto dal file, carattere

i, indice della riga, intero > 0

INIZIO

i = 1

c = leggere_da_file(file)

MENTRE ((c != LINEA) AND (c != FINE_FILE))


```
elemento in posizione i di riga = c
i = i + 1
c = leggere_da_file(file)
fine MENTRE
elemento in posizione i di riga = FINE_STRINGA
FINE
```

Funzione leggere_tastiera_int_verificato:

INPUT:

//nessuno

OUTPUT:

numero, numero inserito dall'utente, intero

LAVORO:

i, indice di scorrimento di input, intero > 0

input, array di caratteri contenente l'input da tastiera,
array di caratteri

c, carattere letto da tastiera, carattere

INIZIO

MAX_INPUT = 2

i = 1

elemento in posizione i di input = leggere_da_tastiera()

MENTRE ((elemento in posizione i di input != LINEA) AND (i
<= MAX_INPUT)

i = i + 1

SE (i = MAX_INPUT)

c = leggere_da_tastiera()

SE (c != LINEA)

```
numero = -infinito  
ALTRIMENTI  
elemento in posizione i di input = leggere_da_tastiera()  
fine SE  
fine MENTRE  
SE (numero != -infinito)  
numero = convertire_ascii_intero(input, MAX_INPUT)  
fine SE  
FINE
```

Funzione convertire_ascii_intero:

INPUT:

numero_char, array di caratteri contenente le cifre di un
numero, array di caratteri

dim, dimensione di numero_char, intero > 0

OUTPUT:

numero, numero convertito, intero

LAVORO:

i, indice nell'array di cifre, intero > 0

esponente, esponente della cifra, Intero >= 0

valore_ascii, valore nella tavola ascii di un carattere,
intero >= 0

cifra, conversione a intero del carattere letto dall'array
di caratteri, intero

potenza, potenza di 10 elevato all'esponente, intero

INIZIO

```

BASE_DECIMALE = 10
ASCII_9 = '9'
ASCII_0 = '0'
numero = 0
i = dim - 1
esponente = 0
SE (elemento in posizione dim - 1 di numero_char = LINEA)
i = i - 1
fine SE
MENTRE ((i >= 0) AND (numero != -infinito))
valore_ascii = elemento in posizione i di numero_char
SE ((valore_ascii <= ASCII_9) AND (valore_ascii >= ASCII_0))
cifra = valore_ascii - ASCII_0
potenza = calcolare_potenza(BASE_DECIMALE, esponente)
numero = numero + (cifra * potenza)
i = i - 1
esponente = esponente + 1
ALTRIMENTI
numero = -infinito
fine SE
fine MENTRE
FINE

```

Funzione calcolare_potenza:

INPUT:

base_numero, base di cui calcolare la potenza, intero >0

esponente, numero di volte per cui moltiplicare la base_numero per se stesso, intero ≥ 0

OUTPUT:

potenza, numero ricevuto in input elevato a potenza, Intero

LAVORO:

i, numero di volte per cui è stata effettuata la potenza, intero, ≥ 0

INIZIO

i = 1

potenza = 1

MENTRE (i \leq esponente)

potenza = potenza * base_numero

i = i + 1

fine MENTRE

FINE

Funzione scegliere_opzione_di_menu:

INPUT:

file_menu, file di testo contenente il menu da stampare, FILE di testo

min_scelta, minimo valore inseribile dall'utente, intero ≥ 0

max_scelta, massimo valore inseribile dall'utente, intero ≥ 0

OUTPUT:

scelta, scelta effettuata dall'utente nei vincoli prestabiliti, intero ≥ 0

INIZIO

ESEGUI

stampare_file_di_testo(file_menu)

scelta = leggere_tastiera_int_verificato()

FINCHÈ ((scelta > max_scelta) OR (scelta < min_scelta))

FINE