MODULO - GESTIRE_PARTITE

Moduli Inclusi:

GESTIRE_PARTITE

GESTIRE_FILE

GESTIRE_MENU

Costanti:

INIZIO = -1

FINE = 1

LANCIO_MINIMO = 1

LANCIO_MASSIMO = 6

LANCIO_OCA = 9

Algoritmi:

Funzione scrivere_nome_giocatore

INPUT:

giocatore, giocatore di cui inserire un carattere del nome, Record giocatore

pos, posizione in cui inserire il carattere del nome, Intero

c, carattere da inserire nella posizione ricevuta in input, Intero

OUTPUT:

giocatore, giocatore di cui è stato aggiornato il campo nome_giocatore

INIZIO

elemento in posizione pos del campo nome_giocatore di giocatore = c

FINE

Funzione leggere_nome_giocatore

INPUT:

giocatore, giocatore di cui leggere un carattere del nome, Record giocatore

pos, posizione in cui leggere il carattere, Intero

OUTPUT:

c, carattere letto nella posizione ricevuta in input

INIZIO

c = elemento in posizione pos del campo nome_giocatore di giocatore

FINE

Funzione scrivere_posizione

INPUT:

giocatore, giocatore di cui scrivere la posizione, Record giocatore

posizione_casella, posizione da inserire, Intero >= 0 <= MAX_PERCORSO

OUTPUT:

giocatore, giocatore di cui è stato aggiornato il campo posizione

INIZIO

elemento del campo posizione di giocatore = posizione_casella

FINE

Funzione leggere_posizione

INPUT:

giocatore, giocatore di cui leggere la posizione, Record giocatore

OUTPUT:

posizione_casella, posizione in cui si trova il giocatore, Intero >= 0 <= MAX_PERCORSO

INIZIO

posizione_casella = elemento del campo posizione di giocatore

FINE

Funzione scrivere_lanci

INPUT:

giocatore, giocatore di cui scrivere il numero di lanci, Record giocatore

lanci_totali, numero di lanci effettuati da inserire, Intero >= 0

OUTPUT:

giocatore, giocatore di cui è stato aggioranto il campo lanci

INIZIO

elemento del campo lanci di giocatore = lanci_totali

FINE

Funzione leggere_lanci

INPUT:

giocatore, giocatore di cui leggere il numero di lanci, Record giocatore

OUTPUT:

lanci_totali, numero di lanci effettuati dal giocatore, Intero >= 0

INIZIO

Lanci_totali = elemento del campo lanci di giocatore

FINE

Funzione scrivere_blocco:

INPUT:

giocatore, giocatore di cui indicare lo stato di blocco, Record giocatore

valore_blocco, valore che indica il numero di turni che il giocatore non può giocare, Intero

OUTPUT:
giocatore, giocatore di cui è stato aggiornato il campo blocco

INIZIO

elemento del campo blocco di giocatore = valore_blocco

FINE

Funzione leggere_blocco

INPUT:

giocatore, giocatore di cui leggere lo stato di blocco, Record giocatore

OUTPUT:

valore_blocco, valore che indica il numero di turni che il giocatore non può giocare, Intero

INIZIO

valore_blocco = elemento del campo blocco di giocatore

FINE

Funzione scrivere_tabellone_percorso

INPUT:

partita, partita di cui aggiornare il campo tabellone_percorso, Record competizione_oca

serpentina, tabellone a serpentina da inserire nella partita, Record tabellone

OUTPUT:

partita, partita di cui è stato aggiornato il campo tabellone_percorso

INIZIO

elemento del campo tabellone_percorso di partita = serpentina

FINE

Funzione leggere_tabellone_percorso

INPUT:

partita, partita di cui leggere il campo tabellone_percorso, Record competizione_oca

OUTPUT:

serpentina, tabellone da leggere, Record tabellone

INIZIO

serpentina = elemento del campo tabellone_percorso di partita

FINE

Funzione scrivere_num_giocatori

INPUT:
partita, partita di cui aggiornare il campo num_giocatori, Record competizione_oca

num_giocatori, numero dei giocatori della partita, Intero, >= MIN_GIOCATORI <= MAX_GIOCATORI

OUTPUT:

partita, partita di cui è stato aggiornato il campo num_giocatori, Record competizione_oca

INIZIO

elemento del campo num_giocatori di partita = num_giocatori

FINE

Funzione leggere_num_giocatori

INPUT:

partita, partita di cui leggere il campo num_giocatori, Record competizione_oca

OUTPUT:

num_giocatori, numero dei giocatori, >= MIN_GIOCATORI <= MAX_GIOCATORI

INIZIO

num_giocatori = elemento del campo num_giocatori di partita

Funzione scrivere_giocatori

INPUT:

partita, partita di cui aggiornare il campo giocatori in posizione pos, Record competizione_oca

pos, pos in cui inserire l'elemento in giocatore, Intero

giocatore, giocatore da inserire nella lista dei giocatori, Record giocatore

OUTPUT:

partita, partita di cui è stato aggiornato il campo giocatori, Record competizione_oca

INIZIO

elemento in posizione pos del campo giocatori di partita = giocatore

FINE

Funzione leggere_giocatori

INPUT:

partita, partita da leggere il campo giocatori in posizione pos, Record competizione_oca

pos, pos in cui leggere l'elemento in giocatori, Intero

OUTPUT:

giocatore, giocatore letto in giocatori in posizione pos, Record giocatore

INIZIO

giocatore = elemento in posizione pos del campo giocatori di partita

FINE

Funzione scrivere_turno

INPUT:

partita, partita di cui aggiornare il campo turno, Record competizione_oca

turno, turno della partita, Intero >= 0

OUTPUT:

partita, partita di cui è stato aggiornato il campo turno, Record competizione_oca

INIZIO

elemento del campo turno di partita = turno

FINE

Funzione leggere_turno

INPUT:

partita, partita di cui leggere il campo turno, Record competizione_oca

OUTPUT:

turno, turno letto dalla partita, Intero >= 0

INIZIO

turno = elemento del campo turno di partita

FINE

Funzione gestire_turno:

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

vincitore, vincitore della partita, record giocatore

OUTPUT:

vincitore, giocatore vincitore della partita, Record giocatore

LAVORO:

num_giocatori, numero di giocatori partecipanti, Intero, >= MIN_GIOCATORI <= MAX_GIOCATORI

turno, turno attualmente in corso, Intero >= 0 < MAX_GIOCATORI

fine, variabile che indica la fine del gioco, Intero

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

blocco_turno, indica se il giocatore è bloccato, Intero

INIZIO

num_giocatori = leggere_num_giocatori(partita_in_corso)

partita_in_corso = fissare_turno(partita_in_corso)

turno = leggere_turno(partita_attuale)

fine = 0

MENTRE (fine != 1)

MENTRE ((turno <= num_giocatori) AND (fine != 1))

partita_in_corso = leggere_giocatori(partita_in_corso, turno,giocatore_attuale)

blocco_turno = leggere_blocco(giocatore_attuale)

SE (blocco_turno >= 0)

fine = gestire_partita_in_corso(partita_in_corso)

SE ((fine != 1) AND (fine != ERRORE_FILE))

partita_in_corso = gestire_turno_generale (partita_in_corso, giocatore_attuale, turno)

num_giocatori = leggere_num_giocatori(partita_in_corso)

SE (num_giocatori = FINE)

```
fine = 1
partita_in_corso = leggere_giocatori (partita_in_corso,
turno, giocatore_attuale)
vincitore = giocatore_attuale
fine SE
ALTRIMENTI SE (fine = 1)
vincitore = scrivere_lanci(vincitore, PARTITA_INTERROTTA)
fine SE
ALTRIMENTI
vincitore = scrivere_lanci(vincitore, ERRORE_FILE)
fine SE
fine SE
ALTRIMENTI
partita_in_corso = gestire_turno_blocco
(partita_in_corso, giocatore_attuale, turno)
fine SE
partita_in_corso = aggiornare_turno (partita_in_corso)
turno = leggere_turno(partita_in_corso)
FINE
turno = 0
fine MENTRE
stampare_vincitore(vincitore)
FINE
```

Funzione fissare_turno

INPUT
partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

num_giocatori, valore che indica il numero di giocatori che partecipano alla partita, Intero >= MIN_GIOCATORI <= MAX_GIOCATORI

turno, indica il giocatore che inizierà il turno, Intero > 0 < num_giocatori

INIZIO

num_giocatori = leggere_num_giocatori (partita_in_corso)

turno = leggere_turno(partita_in_corso)

SE(turno = INIZIO)

turno = generare_numero (num_giocatori + 1) - 1

partita_in_corso = scrivere_turno(partita_in_corso, turno)

fine SE
FINE

Funzione stampare_vincitore

INPUT:

vincitore, vincitore della partita, Record giocatore

OUTPUT:
//... Nessuno

LAVORO:

validita_vittoria, indica se il giocatore ha vinto o meno la partita, Intero

nome_vincitore, nome del giocatore che ha vinto la partita, array di caratteri

i, indice dell'array del nome giocatore, intero

INZIO

AVVISO_VITTORIA = " Hai vinto giocatore "

validita_vittoria = leggere_lanci (vincitore)

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

i = 1

MENTRE (i < NOME)

elemento in posizione i di nome_vincitore = leggere_nome_giocatore(vincitore, i);

i = i + 1;

fine MENTRE

stampare_a_video(LINEA)

stampare_a_video(AVVISO_VITTORIA, nome_vincitore)

stampare_a_video(LINEA)

fine SE

FINE

Funzione gestire_turno_blocco

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

turno, turno attualmente in corso, Intero >= 0 < MAX_GIOCATORI

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO

posizione_attuale, posizione del giocatore, Intero >= 0 <= MAX_PERCORSO

blocco_turno, indica se il giocatore è bloccato, Intero

tabellone_percorso, tabellone della partita, Record tabellone

casella, casella sulla quale il giocatore è attualmente posizionato, Intero >= CASELLA_VUOTA <= CASELLA_SCHELETRO

INIZIO

posizione_attuale = leggere_posizione(giocatore_attuale)

blocco_turno = leggere_blocco(giocatore_attuale)

tabellone_percorso = leggere_tabellone_percorso(partita_attuale, tabellone_percorso)

```
casella = leggere_percorso(tabellone_percorso,
posizione_attuale)

SE(casella = POSIZIONE_PRIGIONE)

visualizzare_partita(partita_in_corso,
posizione_attuale)

giocatore_attuale =
liberare_prigione_lancio(giocatore_attuale)

ALTRIMENTI

blocco_turno = blocco_turno + 1

giocatore_attuale = scrivere_blocco(giocatore_attuale,
blocco_turno)

fine SE

partita_in_corso= scrivere_giocatori(partita_in_corso,
turno, giocatore_attuale)

FINE
```

Funzione aggiornare_turno

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:
turno, turno aggiornato della partita, Intero >= 0

LAVORO:

num_giocatori, numero dei giocatori della partita, intero >0 <= MAX_GIOCATORI

```
INIZIO
num_giocatori = 0
turno = 0
turno = leggere_turno(partita_in_corso)
num_giocatori = leggere_num_giocatori(partita_in_corso)
turno = turno + 1
SE(turno < num_giocatori)
partita_in_corso =
scrivere_turno(partita_in_corso,turno)
ALTRIMENTI
partita_in_corso = scrivere_turno(partita_in_corso,0)
fine SE
FINE
```

Funzione gestire_turno_generale

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

turno, turno in corso della partita, Intero >= 0

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

fine, variabile che indica la fine del gioco, Intero

tabellone_percorso, tabellone della partita, Record tabellone

dimensione, numero di caselle del tabellone, Intero >= MIN_PERCORSO <= MAX_PERCORSO

primo_lancio, lancio di un dado effettuato, Intero, >= LANCIO_MINIMO<= LANCIO_MASSIMO

secondo_lancio, lancio di un dado effettuato, Intero, >= LANCIO_MINIMO<= LANCIO_MASSIMO

lancio, somma dei due lanci di dadi, Intero

posizione_attuale, posizione del giocatore, Intero, >= 0 <= MAX_PERCORSO

lanci, numero di tiri effettuati dal giocatore, Intero >= 0

indietro, variabile che indica lo spostamento effettuato all'indietro dal giocatore, Intero > 0

casella, valore che rappresenta la casella, intero > 0

blocco_turno, indica se il giocatore è bloccato,Intero

pos_supporto, variabile di appoggio, Intero

INIZIO

LANCIO = "Il numero di casella di cui spostarsi e': "

fine = 0

pos_supporto = 0

blocco_turno = 0

casella = 0

posizione_attuale = 0

```
lanci = 0
lancio = 0
secondo_lancio = 0
primo_lancio = 0
dimensione = 0
tabellone_percorso =
leggere_tabellone_percorso(partita_in_corso,
tabellone_percorso)
dimensione = leggere_dimensione(tabellone_percorso)
primo_lancio = generare_numero(LANCIO_MASSIMO + 1)
secondo_lancio = generare_numero (LANCIO_MASSIMO + 1)
lancio = primo_lancio + secondo_lancio
stampare_a_video(LINEA, LANCIO, lancio, LINEA)
posizione_attuale = leggere_posizione(giocatore_attuale)
posizione_attuale = posizione_attuale + lancio
giocatore_attuale =
scrivere_posizione(giocatore_attuale, posizione_attuale)
lanci = leggere_lanci(giocatore_attuale)
lanci = lanci + 1
giocatore_attuale = scrivere_lanci (giocatore_attuale,
lanci)
indietro = 0
casella = leggere_percorso(tabellone_percorso,
posizione_attuale)
ESEGUI
SE (poizione_attuale > dimensione)
```

```
posizione_attuale =
gestire_tabellone_lancio_vittoria(dimensione,
posizione_attuale, lancio)

giocatore_attuale =
scrivere_posizione(giocatore_attuale, posizione_attuale)

indietro = posizione_attuale - dimensione + 1

casella = leggere_percorso(tabellone_percorso,
posizione_attuale)

ALTRIMENTI SE (posizione_attuale = dimensione - 1)

fine = 1

partita_in_corso = scrivere_giocatori(partita_in_corso,
FINE)

ALTRIMENTI SE (casella != CASELLA_VUOTA)

partita_in_corso = scrivere_ giocatori(partita_in_corso,
turno, giocatore_attuale)

SE (indietro < 0)

visualizzare_partita(partita_in_corso,
posizione_attuale)

pos_supporto = pos_attuale

primo_lancio = 0

secondo_lancio = indietro

partita_in_corso =
gestire_caselle_speciali(partita_in_corso, primo_lancio,
secondo_lancio, casella)

giocatore_attuale = leggere_giocatori(partita_in_corso,
turno,giocatore_attuale)

posizione_attuale = leggere_posizione(giocatore_attuale)

indietro = posizione_attuale - pos_supporto
```

```
ALTRIMENTI
visualizzare_partita(partita_in_corso,
posizione_attuale)
casella = leggere_percorso(tabellone_percorso,
posizione_attuale)
partita_in_corso =
gestire_caselle_speciali(partita_in_corso, primo_lancio,
secondo_lancioo, casella)
giocatore_attuale = leggere_ giocatori(partita_in_corso,
turno,giocatore_attuale)
posizione_attuale = leggere_posizione(giocatore_attuale)
fine SE
casella = leggere_percorso(tabellone_percorso,
posizione_attuale)
blocco_turno = leggere_blocco(giocatore_attuale)
fine SE
FINCHÈ ((casella != CASELLA_NORMALE) AND (blocco_turno >=
0) AND (fine != 1) AND (posizione_attuale != -1))
partita_in_corso = scrivere_giocatori(partita_in_corso,
turno, giocatore_attuale)
FINE
```

Funzione gestire_caselle_speciali

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

primo_lancio, primo dado lanciato, Intero >= MIN_LANCIO <= MAX_LANCIO

secondo_lancio, secondo dado lanciato, Intero >= MIN_LANCIO <= MAX_LANCIO

casella_speciale, casella speciale sulla quale si trova il giocatore, Intero >= POSIZIONE_OCA <= POSIZIONE_SCHELETRO

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

lancio, somma dei due lanci di dadi, Intero

INIZIO

lancio = primo_lancio + secondo_lancio

SE (casella_speciale = POSIZIONE_OCA)

partita_in_corso = gestire_casella_oca(partita_in_corso, primo_lancio, secondo_lancio)

ALTRIMENTI SE (casella_speciale = POSIZIONE _PONTE)

partita_in_corso = gestire_casella_ponte(partita_in_corso, lancio)

ALTRIMENTI SE (casella_speciale = POSIZIONE _LOCANDA)

partita_in_corso = gestire_casella_locanda(partita_in_corso)

ALTRIMENTI SE ((casella_speciale = POSIZIONE _POZZO) OR (casella_speciale = POSIZIONE _PRIGIONE))

partita_in_corso = gestire_casella_blocco(partita_in_corso)

ALTRIMENTI SE (casella_speciale = POSIZIONE _LABIRINTO)

partita_in_corso =
gestire_casella_labirinto(partita_in_corso)

ALTRIMENTI SE (casella_speciale = POSIZIONE _SCHELETRO)

partita_in_corso =
gestire_casella_scheletro(partita_in_corso)

FINE

Funzione gestire_casella_oca

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

primo_lancio, primo dado lanciato, Intero >= MIN_LANCIO <= MAX_LANCIO

secondo_lancio, secondo dado lanciato, Intero >= MIN_LANCIO <= MAX_LANCIO

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

turno, turno attualmente in corso, Intero, >= 0 < MAX_GIOCATORI

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

posizione_attuale, posizione del giocatore, Intero, >= 0 <= MAX_PERCORSO

```
lanci, numero di tiri effettuati dal giocatore, Intero, >=
0


INIZIO
AVVISO_OCA_1 = "Sei finito su un oca."
AVVISO _OCA_2 = "Avanza ancora."
POS_SPECIALE_1 = 25
POS_SPECIALE_2 = 52
LANCIO_SPECIALE_1 = 3
LANCIO_SPECIALE_2 = 6
stampare_a_video(LINEA)
stampare_a_video(AVVISO _OCA_1)
stampare_a_video(LINEA)
stampare_a_video(AVVISO _OCA_2)
stampare_a_video(LINEA)
turno = leggere_turno(partita_in_corso)
giocatore_attuale = leggere_ giocatori(partita_in_corso,
turno, giocatore_attuale)
lanci = leggere_lanci(giocatore_attuale)
posizione_attuale = leggere_posizione(giocatore_attuale)
SE (lanci = 1)
SE (((primo_lancio = LANCIO_SPECIALE_1) AND
(secondo_lancio = LANCIO_SPECIALE_2)) OR ((secondo_lancio
= LANCIO_SPECIALE_1) AND (primo_lancio =
LANCIO_SPECIALE_2)))
posizione_attuale = POS_SPECIALE_1
```

ALTRIMENTI

posizione_attuale = POS_SPECIALE_2

fine SE

ALTRIMENTI

posizione_attuale = posizione_attuale + (primo_lancio + secondo_lancio)

fine ALTRIMENTI

giocatore_attuale = scrivere_posizione(giocatore_attuale, posizione_attuale)

partita_in corso= scrivere_giocatori(partita_in_corso, turno, giocatore_attuale)
FINE

Funzione gestire_casella_ponte

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

lancio, lancio dei dadi effettuato, Intero >= MIN_LANCIO <= lancio <= MAX_LANCIO

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

turno, turno attualmente in corso, Intero >= 0 < MAX_GIOCATORI

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

posizione_attuale, posizione del giocatore, Intero >= 0 <= MAX_PERCORSO

INIZIO

AVVISO_POSTA = "Sei finito sulla casella ponte"

stampare_a_video(AVVISO_POSTA)

turno = leggere_turno(partita_in_corso)

giocatore_attuale = leggere_giocatori(partita_in_corso, turno,giocatore_attuale)

posizione_attuale = leggere_posizione(giocatore_attuale)

posizione_attuale = posizione_attuale + lancio

giocatore_attuale = scrivere_posizione(giocatore_attuale, posizione_attuale)

partita_in_corso = scrivere_giocatori(partita_in_corso, turno, giocatore_attuale)

FINE

Funzione gestire_casella_locanda

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

turno, turno attualmente in corso, Intero >= 0 turno < MAX_GIOCATORI

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

INIZIO

AVVISO_LOCANDA = "Sei arrivato sulla casella locanda e sei bloccato!"

FUNZIONE_LOCANDA = -3

stampare_a_video(LINEA)

stampare_a_video(AVVISO_LOCANDA)

stampare_a_video(LINEA)

turno = leggere_turno(partita_in_corso)

giocatore_attuale= leggere_giocatori(partita_in_corso, turno,giocatore_attuale)

giocatore_attuale = scrivere_blocco(giocatore_attuale, FUNZIONE_LOCANDA)

partita_in_corso = scrivere_giocatori(partita_in_corso, turno, giocatore_attuale)

FINE

Funzione liberare_giocatore

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

posizione_attuale, posizione attuale del giocatore, >= 0 <= MAX_PERCORSO

turno, turno attualmente in corso, Intero >= 0 turno < MAX_GIOCATORI

num_giocatori, numero di giocatori partecipanti, >= MIN_GIOCATORI <= MAX_GIOCATORI

i, indice nella lista dei giocatori, Intero >= 0

giocatore_controllo, giocatore controllato per verificare se è bloccato, Record giocatore

blocco_verifica, indica il blocco del giocatore, Intero

posizione_verifica, posizione del giocatore controllata, >= 0 <= MAX_PERCORSO

fine_ciclo, indica lo sblocco di un giocatore, Intero

INIZIO

LIBERA = "Sei bloccato! Ma hai liberato l’altro giocatore"

turno = leggere_turno(partita_in_corso)

giocatore_attuale = leggere_giocatore(partita_in_corso,turno,giocatore_attuale)

posizione_attuale = leggere_posizione(giocatore_attuale)

```
num_giocatori = leggere_num_giocatori(partita_in_corso)
i = 1
fine_ciclo = 0
MENTRE ((i <= num_giocatori) AND (fine_ciclo = 0))
SE (i != turno)
giocatore_controllo =
leggere_giocatori(partita_in_corso,
i,giocatore_controllo)
blocco_verifica = leggere_blocco(giocatore_controllo)
SE (blocco_verifica < 0)
posizione_verifica =
leggere_posizione(giocatore_controllo)
SE (posizione_verifica = pos_attuale)
scrivere_blocco(giocatore_controllo, 0)
fine_ciclo = 1
partita_in_corso = scrivere_giocatori(partita_in_corso,
i, giocatore_controllo)
stampare_a_video(LINEA)
stampare_a_video(LIBERA)
stampare_a_video(LINEA)
fine SE
fine SE
fine SE
i = i + 1
fine MENTRE
FINE
```

Funzione gestire_casella_blocco:

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

turno, turno attualmente in gioco, Intero >= 0 < MAX_GIOCATORI

giocatore_turno, giocatore attualmente in gioco, Record giocatore

INIZIO

AVVISO_BLOCCO = "Sei bloccato"

TURNI_BLOCCO = -INFINITO

stampare_a_video(LINEA)

stampare_a_video(AVVISO_BLOCCO)

stampare_a_video(LINEA)

turno = leggere_turno(partita_in_corso)

giocatore_turno = leggere_giocatori(partita_in_corso, turno, giocatore_turno)

giocatore_turno = scrivere_blocco(giocatore_turno, TURNI_BLOCCO)

partita_in_corso = scrivere_giocatori(partita_in_corso, turno, giocatore_turno)

partita_in_corso = liberare_giocatore(partita_in_corso)

FINE

Funzione gestire_casella_labirinto

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

tabellone, tabellone della partita, Record tabellone

dimensione, dimensione del tabellone, >= MIN_PERCORSO <= MAX_PERCORSO

ritorno, casella a cui il giocatore è mandato, Intero > 0 < MAX_PERCORSO

turno, turno attualmente in gioco, Intero >= 0 < MAX_GIOCATORI

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

posizione_attuale, posizione attuale del giocatore, >= 0 <= MAX_PERCORSO

INIZIO

```
AVVISO_LABIRINTO = "Oh no! Sei sul labirinto torna alla:
"
FUNZIONE_LABIRINTO = 33
stampare_a_video(LINEA)
stampare_a_video(AVVISO_LABIRINTO)
stampare_a_video(FUNZIONE_LABIRINTO)
stampare_a_video(LINEA)
tabellone = leggere_tabellone_percorso(partita_in_corso,
tabellone)
dimensione = leggere_dimensione(tabellone)
ritorno = calcolare_proporzione (FUNZIONE_LABIRINTO,
dimensione, MAX_PERCORSO)
turno = leggere_turno(partita_in_corso)
giocatore_attuale = leggere_giocatori(partita_in_corso,
turno, giocatore_attuale)
posizione_attuale = leggere_posizione(giocatore_attuale)
posizione_attuale = ritorno
giocatore_attuale =
scrivere_posizione(giocatore_attuale, posizione_attuale)
partita_in_corso= scrivere_giocatori(partita_in_corso,
turno, giocatore_attuale)
FINE
```

Funzione gestire_casella_scheletro

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

turno, turno attualmente in gioco, Intero >= 0 < MAX_GIOCATORI

giocatore_attuale, giocatore attualmente in gioco, Record giocatore

posizione_attuale, posizione attuale del giocatore, >= 0 <= MAX_PERCORSO

INIZIO

AVVISO_SCHELETRO = "oh no! Sei sullo scheletro torna alla: "

FUNZIONE_SCHELETRO = 0

stampare_a_video(LINEA)

stampare_a_video(AVVISO_SCHELETRO)

stampare_a_video(FUNZIONE_SCHELETRO)

stampare_a_video(LINEA)

turno = leggere_turno(partita_in_corso)

giocatore_attuale = leggere_giocatori(partita_in_corso, turno,giocatore_attuale)

posizione_attuale = leggere_posizione(giocatore_attuale)

posizione_attuale = FUNZIONE_SCHELETRO - 1

giocatore_attuale =
scrivere_posizione(giocatore_attuale, posizione_attuale)

partita_in_corso = scrivere_giocatori(partita_in_corso,
turno, giocatore_attuale)

FINE

Funzione gestire_tabellone_lancio_vittoria

INPUT:

num_caselle, numero di caselle del tabellone della partita,
Intero >= MIN_PERCORSO <= MAX_PERCORSO

posizione_attuale, posizione attuale del giocatore dopo il
lancio dei dadi, Intero

lancio, lancio dei dadi effettuato, >= MIN_LANCIO <=
MAX_LANCIO

OUTPUT:

posizione_attuale, posizione attuale del giocatore dopo il
lancio dei dadi, Intero

LAVORO:

differenza, numero di caselle oltre la fine della mappa,
Intero

INIZIO

differenza = posizione_attuale - num_caselle

posizione_attuale = ((num_caselle - 1) - differenza) -1

FINE

Funzione liberare_prigione_lancio

```
INPUT:
giocatore_prigione, giocatore bloccato in prigione, Record giocatore
OUTPUT:
giocatore_prigione, giocatore aggiornato, Record giocatore
LAVORO:
lancio, lancio effettuato dal giocatore bloccato, Intero >= MIN_LANCIO <= MAX_LANCIO

INIZIO
AVVISO_LIBERO = "Bravo! sei libero lancio fortunato di: "
AVVISO_BLOCCO = "Peccato! Lancio sfortunato ritenta!"
LIBERA_1 = 5
LIBERA_2 = 7
lancio = generare_numero(LANCIO_MASSIMO + 1)
SE ((lancio = LIBERA_1) OR (lancio = LIBERA_2))
giocatore_prigione = scrivere_blocco(giocatore_prigione, 0)
stampare_a_video(LINEA)
stampare_a_video(AVVISO_LIBERO)
stampare_a_video(lancio)
stampare_a_video(LINEA)
ALTRIMENTI
stampare_a_video(AVVISO_BLOCCO)
stampare_a_video(LINEA)
```

FINE

Funzione configurare_partita_classica

INPUT:

partita, partita classica, Record competizione_oca

OUTPUT:

partita, partita classica, Record competizione_oca

INIZIO

partita = configurare_partita(partita);

FINE

Funzione configurare_partita

INPUT:

partita, partita classica da configurare, Record competizione_oca

OUTPUT:

partita, partita classica da configurare, Record competizione_oca

LAVORO:

scelta, scelta effettuata dall'utente, Intero

tabellone, tabellone del gioco, record tabellone

```
INIZIO

PERCORSO_STANDARD = 90
GIOCATORI_STANDARD = 4

tabellone = scrivere_dimensione(tabellone,
PERCORSO_STANDARD)

tabellone = generare_percorso(&tabellone)

partita = scrivere_tabellone_percorso(partita, tabellone)

partita = scrivere_num_giocatori(partita,
GIOCATORI_STANDARD)

partita = configurare_giocatori(partita,
GIOCATORI_STANDARD)

loading_bar()

partita = scrivere_turno(partita, INIZIO);

FINE
```

Funzione configurare_partita_personalizzata

INPUT:

partita, partita personalizzata da configurare, Record competizione_oca

OUTPUT:

partita, partita personalizzata da configurare, Record competizione_oca

```
INIZIO

partita = configurare_partita_p(partita)
```

```
FINE
```

```
Funzione configurare_partita_p
INPUT:
partita, partita personalizzata da configurare, Record competizione_oca
OUTPUT:
partita, partita personalizzata da configurare, Record competizione_oca
LAVORO
Tabellone, tabellone del gioco, record tabellone
dimensione, dimensione del tabellone, intero
num_giocatori, numero dei giocatori, intero

INIZIO
AVVISO_IMPOSTAZIONI_2 = “---PARTITA PERSONALIZZATA---”
stampa_a_video(AVVISO_IMPOSTAZIONI_2)
AVVISO_CASELLE = "Inserire la dimensione delle caselle minimo 50 massimo 90: "
AVVISO_GIOCATORI ="Inserire il numero di giocatori minimo 2 massimo 4: "
ESEGUI
stampa_a_video(AVVISO_CASELLE)
dimensione = leggere_tastiera_int_verificato()
```

```
FINCHÉ((dimensione < MIN_PERCORSO) OR (dimensione > MAX_PERCORSO))
ESEGUI
zstampa_a_video(AVVISO_GIOCATORI)
num_giocatori = leggere_tastiera_int_verificato()
FINCHÉ ((num_giocatori < MIN_GIOCATORI) OR (num_giocatori > MAX_GIOCATORI))
tabellone = scrivere_dimensione(tabellone, dimensione)
tabellone = generare_percorso(tabellone)
tabellone = scrivere_tabellone_percorso(partita, tabellone)
partita = scrivere_num_giocatori(partita, num_giocatori)
partita = configurare_giocatori(partita, num_giocatori)
loading_bar()
partita = scrivere_turno(partita, INIZIO)
FINE
```

Funzione verifica_numero

INPUT:

c, carattere da verificare, carattere

OUTPUT:

c, carattere verificato, carattere

```
INIZIO
c = (c >= '0' AND c <= '9')
```

FINE

Funzione verifica_carattere_speciale

INPUT:

c, carattere da verificare, carattere

OUTPUT:

c, carattere verificato, carattere

INIZIO

c != ((c >= ‘a’ AND c <= ‘z’) OR (c >= ‘A’ AND c <= ‘Z’))

FINE

Funzione configurare_giocatori

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

num_giocatori, numero di giocatori partecipanti, Intero >= MIN_GIOCATORI <= MAX_GIOCATORI

OUTPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

LAVORO:

i, indice nella lista dei giocatori, Intero >= 0

giocatore, giocatore da configurare, Record giocatore

```
j, indice nella stringa del nome del giocatore, Intero >= 0
c, carattere da inserire nel nome del giocatore, Carattere
fine_input, indica se il nome inserito ha spazi vuoti, intero

INIZIO
A_CAPO = '\n'
i = 1
MENTRE (i <= num_giocatori)
giocatore = scrivere_posizione(giocatore, -1)
giocatore = scrivere_lanci(giocatore, 0)
giocatore = scrivere_blocco(giocatore, 0)
ESEGUI
c = FINE_STRINGA
fine_input = 0
j = 1
MENTRE ((j < NOME - 1) AND (fine_input != 1)
stampare_a_video(“NOME DA INSERIRE DI 5 CARATTERI”)
stampare_a_video(“Inserire il nome del giocatore : ”i+1)
c = leggere_da_tastiera
SE(c = A_CAPO)
fine_input = 1
ALTRIMENTI
giocatore = scrivere_nome_giocatore(giocatore, j, c)
```

```
fine SE
j = j + 1
fine MENTRE
giocatore = scrivere_nome_giocatore(giocatore, j,
FINE_STRINGA)
FINCHÈ ((j != LUNGH_NOME) || (fine_input = 1))
partita_in_corso = scrivere_ giocatori(partita_in_corso,
i, giocatore)
i = i + 1
fine MENTRE
FINE
```

Funzione generare_numero

INPUT:

numero, numero da generare, intero >0

OUTPUT:

num_casuale, numero generato casualmente, intero

LAVORO:

t, valore che indica l'orario attuale, intero

```
INIZIO
t = ottenere_tempo_corrente()
inizializzare_generatore_numeri_casuali(t)
num_casuale = 0
ESEGUI
```

```
num_casuale = generare_numero_casuale_fino_a(numero)
FINCHÈ (num_casuale = 0)
FINE
```

Funzione visualizzare_partita

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

posizione_attuale, posizione attuale del giocatore disputante, >= 0 <= MAX_PERCORSO

OUTPUT:

//...Nessuno

LAVORO:

tabellone_percorso, tabellone della partita in corso, Record tabellone

turno, turno attualmente in gioco, Intero >= 0 < MAX_GIOCATORI

```
INIZIO
STAMPA_TAB = "Tabellone: "
tabellone_percorso = leggere_tabellone_percorso(partita_in_corso, tabellone_percorso)
turno = leggere_turno(partita_in_corso)
stampare_a_video(STAMPA_TAB)
stampare_percorso(tabellone_percorso,posizione_attuale)
```

partita_in_corso = stampare_posizioni(partita_in_corso, turno)

FINE

Funzione stampare_posizioni

INPUT:

partita_in_corso, partita in corso di svolgimento, Record competizione_oca

turno, turno attualmente di gioco, Intero >= 0 < MAX_GIOCATORI

OUTPUT:

//…Nessuno

LAVORO:

tabellone_percorso, tabellone della partita in corso, Record tabellone

giocatore_indice, giocatore attualmente in gioco, Record giocatore

i, indice di scorrimento, Intero

nome_giocatore, nome del giocatore che sta effettuando il turno nel gioco, array di caratteri

num_giocatori, numero dei giocatori, intero

contatore_turno, contatore del turno in corso, intero

j, indice nella stringa del nome del giocatore, Intero >= 0

turno, turno del gioco in corso, intero >= 0

INIZIO

```
AVVISO_POSIZIONE_1 = " La posizione del giocatore "

AVVISO_POSIZIONE_2 = " è "

Tabellone_percorso =
leggere_tabellone_percorso(partita_in_corso,tabellone_p
ercoso)

num_giocatori = leggere_num_giocatori(partita_in_corso)

contatore_turno = aggiornare_turno(partita_in_corso)

stampare_a_video(contatore_turno)

i = 1

MENTRE (i < num_giocatori)

Giocatore_indice = leggere_giocatori(partita_in_corso, i,
giocatore_indice)

pos = leggere_posizione(giocatore_indice)

j = 1

MENTRE (j < NOME)

elemento in posizione j di nome_giocatore =
leggere_nome_giocatore(giocatore_indice,j)

j = j +  1

fine MENTRE

SE(i = turno)

stampare_a_video(COLORE_ARANCIONE, AVVISO_POSIZIONE_1,
RESET_COLORE)

stampare_a_video(COLORE_ARANCIONE, nome_giocatore,
RESET_COLORE)

stampare_a_video(COLORE_ARANCIONE, AVVISO_POSIZIONE_2,
RESET_COLORE)

stampare_a_video(COLORE_ARANCIONE, pos+1, RESET_COLORE)
```

```
stampare_a_video(LINEA)

ALTRIMENTI
stampare_a_video(AVVISO_POSIZIONE_1)
stampare_a_video(nome_giocatore)
stampare_a_video(STAMPA_POSIZIONE_2)
stampare_a_video(pos + 1)
stampare_a_video(LINEA)
fine SE
i = i + 1
fine MENTRE
stampare_a_video(LINEA)
FINE

Funzione loading_bar
INPUT
//..Nessuno
OUTPUT
//nessuno
LAVORO:
totale, il valore totale per il completamento della barra di caricamento, intero
barra_lunghezza, la lunghezza della barra di caricamento, intero
velocita, La velocità di avanzamento della barra, intero
```

```
progresso, Il valore di progresso corrente, intero
i, indice di scorrimento sul totale, intero
percentuale, la percentuale di completamento, decimale
larghezza_riempita, la larghezza riempita della barra, intero
j, indice di scorrimento per la larghezza_riempita, intero
k, indice di scorrimento per la barra_lunghezza, intero

INIZIO
totale = 100
barra_lunghezza = 50
velocita = 1000 (microsecondi)
progresso = 0
i = 1
MENTRE (i <= totale)
percentuale = progresso / totale
larghezza_riempita = barra_lunghezza * percentuale
j = 1
MENTRE (j <= larghezza_riempita)
j = j +1
fine MENTRE
k = larghezza_riempita
MENTRE ( k < barra_lunghezza - 1)
k = k + 1
fine MENTRE
```

```
progresso = progresso + 1
velocita = 1000
i = i + 1
fine MENTRE
stampa_a_video(“Loading Completo!”)
FINE
```