

CASI DI TEST MODULO - GESTIRE_PERCORSO

Funzione `calcolare_proporzione`

Caso di test 1:

Descrizione: Valori positivi non nulli per gli elementi della proporzione.

Input: 5, 3, 2

Output Atteso: 3

Caso di test 2:

Descrizione: Valori negativi non nulli per gli elementi della proporzione.

Input: -2, -3, -4

Output Atteso: 2

Caso di test 3:

Descrizione: Tutti gli elementi della proporzione sono nulli. Input: 0, 0, 0

Output Atteso: 0

Caso di test 4:

Descrizione: Il primo elemento è nullo, il secondo e il quarto elemento sono non nulli.

Input: 0, 1, 5

Output Atteso: 0

Funzione `inserire_caselle_oca`:

Caso di test 1:

Descrizione: La board di gioco è composta da 55 caselle. Tutte le caselle sono inizializzate a 0. Vogliamo verificare che le caselle "goose" vengano posizionate correttamente ogni 9 caselle.

Input: Una board di gioco con 55 caselle, tutte impostate a 0.

Output atteso: Le caselle "goose" vengono posizionate ogni 9 caselle, ottenendo un board con 5 caselle "goose", mentre le rimanenti caselle rimangono a 0.

Caso di test 2:

Descrizione: La board di gioco è composta da 85 caselle. Tutte le caselle sono inizializzate a 0. Vogliamo verificare che le caselle "goose" vengano posizionate correttamente ogni 9 caselle.

Input: Una board di gioco con 85 caselle, tutte impostate a 0.

Output atteso: Le caselle "goose" vengono posizionate ogni 9 caselle, ottenendo una board con 10 caselle "goose", mentre le rimanenti caselle rimangono a 0.

Caso di test 3:

Descrizione: La board di gioco è composta da 95 caselle. Tutte le caselle sono inizializzate a 0. Vogliamo verificare che le caselle "goose" vengano posizionate correttamente ogni 9 caselle.

Input: Una board di gioco con 95 caselle, tutte impostate a 0.

Output atteso: Le caselle "goose" vengono posizionate ogni 9 caselle, ottenendo una board con 11 caselle "goose", mentre le rimanenti caselle rimangono a 0.

Caso di test 4:

Descrizione: La board di gioco è composta da 70 caselle. Tutte le caselle sono inizializzate a 0. Vogliamo verificare che le caselle "goose" vengano posizionate correttamente ogni 9 caselle.

Input: Una board di gioco con 70 caselle, tutte impostate a 0.

Output atteso: Le caselle "goose" vengono posizionate ogni 9 caselle, ottenendo una board con 7 caselle "goose", mentre le rimanenti caselle rimangono a 0.

Funzione stabilire_percorso:

Caso di test 1:

Descrizione: Si desidera creare una mappa di gioco con il numero minimo consentito di caselle (50). La mappa deve essere completamente vuota.

Input: Una game board con il numero minimo di caselle (50).

Output atteso: La game board viene restituita con il campo "tabellone_percorso" inizializzato con 50 caselle impostate a 0 (casella normale).

Caso di test 2:

Descrizione: Si desidera creare una mappa di gioco con il numero massimo consentito di caselle (90). La mappa deve essere completamente vuota.

Input: Una game board con il numero massimo di caselle (90).

Output atteso: La game board viene restituita con il campo "tabellone_percorso" inizializzato con 90 caselle impostate a 0 (casella normale).

Caso di test 3:

Descrizione: Si desidera creare una mappa di gioco con un numero di caselle compreso tra il minimo e il massimo consentito (70). La mappa deve essere completamente vuota.

Input: Una game board con un numero di caselle pari a 70.

Output atteso: La game board viene restituita con il campo "tabellone_percoso" inizializzato con 70 caselle impostate a 0 (casella normale).

Caso di test 4:

Descrizione: Si desidera creare una mappa di gioco con un numero negativo di caselle (-10). Nonostante il valore negativo, la funzione dovrebbe inizializzare correttamente la mappa.

Input: Una game board con un numero di caselle pari a -10.

Output atteso: La game board viene restituita con il campo "tabellone_percorso" inizializzato con 0 caselle.

Funzione stampare_riga:

Caso di test 1:

Descrizione: Verifica se la funzione incrementa correttamente la posizione quando la posizione corrente è minore o uguale al numero totale di caselle del tabellone e il numero di elementi stampati nella riga è inferiore al numero di caselle desiderate per riga.

Input: numero_caselle = 10, elementi_riga_corrente = 0, caselle_per_riga = 5, posizione_corrente = 1, posizione_turno = 3

Output atteso: La posizione successiva dopo aver stampato l'elemento corrente dovrebbe essere `posizione_corrente = 5`.

Caso di test 2:

Descrizione: Verifica se la funzione incrementa correttamente la posizione quando la posizione corrente è minore o uguale al numero totale di caselle del tabellone e il numero di elementi stampati nella riga è inferiore al numero di caselle desiderate per riga.

Input: `numero_caselle = 20`, `elementi_riga_corrente = 3`, `caselle_per_riga = 4`, `posizione_corrente = 7`, `posizione_turno = 10`

Output atteso: La posizione successiva dopo aver stampato l'elemento corrente dovrebbe essere `posizione_corrente = 8`.

Caso di test 3:

Descrizione: Verifica se la funzione non incrementa la posizione quando la posizione corrente supera il numero totale di caselle del tabellone.

Input: `numero_caselle = 5`, `elementi_riga_corrente = 4`, `caselle_per_riga = 5`, `posizione_corrente = 6`, `posizione_turno = 2` Output atteso: La posizione dovrebbe rimanere uguale a `posizione_corrente = 6` poiché supera il numero totale di caselle disponibili.

Funzione `stampare_riga_invertita`

Caso di test 1:

Descrizione: Verifica se la funzione stampa correttamente gli elementi quando il numero di elementi stampati nella riga è diverso dal numero di caselle mancanti per completare la riga.

Input: numero_caselle = 10, elementi_riga_corrente = 2,
caselle_per_riga = 4, posizione_corrente = 5,
posizione_turno = 3

Output atteso: La posizione successiva dopo aver
stampato l'elemento corrente dovrebbe essere
posizione_corrente = 8, e il numero di elementi
stampati nella riga dovrebbe essere uguale al numero di
caselle mancanti.

Caso di test 2:

Descrizione: Verifica se la funzione stampa
correttamente gli elementi quando il numero di elementi
stampati nella riga è diverso dal numero di caselle
mancanti per completare la riga.

Input: numero_caselle = 15, elementi_riga_corrente = 5,
caselle_per_riga = 3, posizione_corrente = 12,
posizione_turno = 8

Output atteso: La posizione successiva dopo aver
stampato l'elemento corrente dovrebbe essere
posizione_corrente = 16, e il numero di elementi
stampati nella riga dovrebbe essere uguale al numero di
caselle mancanti.

Caso di test 3:

Descrizione: Verifica se la funzione stampa
correttamente gli elementi quando il numero di elementi
stampati nella riga è uguale al numero di caselle
mancanti per completare la riga.

Input: numero_caselle = 8, elementi_riga_corrente = 2,
caselle_per_riga = 6, posizione_corrente = 4,
posizione_turno = 6

Output atteso: La posizione successiva dopo aver
stampato l'elemento corrente dovrebbe essere
posizione_corrente = 8.

Funzione stampare_riga_minima:

Caso di test 1:

: Verifica se la funzione stampa correttamente gli elementi quando il numero di elementi stampati nella riga è diverso dal numero di caselle da stampare per riga.

Input: numero_elementi_riga = 3, caselle_per_riga = 5, posizione_corrente = 9, posizione_turno = 6

Output atteso: La posizione successiva dopo aver stampato l'elemento corrente dovrebbe essere posizione_corrente = 14, e il numero di elementi stampati nella riga dovrebbe essere uguale al numero di caselle da stampare per riga.

Caso di test 2:

Descrizione: Verifica se la funzione stampa correttamente gli elementi quando il numero di elementi stampati nella riga è diverso dal numero di caselle da stampare per riga.

Input: numero_elementi_riga = 2, caselle_per_riga = 4, posizione_corrente = 6, posizione_turno = 2

Output atteso: La posizione successiva dopo aver stampato l'elemento corrente dovrebbe essere posizione_corrente = 11, e il numero di elementi stampati nella riga dovrebbe essere uguale al numero di caselle da stampare per riga.