

[V]*Es1. Scrivere un metodo che prenda in input due array A e B di int di lunghezza n ed m rispettivamente, e restituisca una matrice di dimensione nxm di int contenente il prodotto AxB se in A tutti gli elementi sono distinti e la matrice identità altrimenti;

[V]*Es2. Data una matrice A di stringhe, restituire un boolean che indica se esiste una colonna in cui siano presenti due stringhe l'una l'inversa dell'altra;

[V]*Es3. Data la matrice A[7][10] di float restituire l'indice di una riga il cui elemento massimo è strettamente minore dei valori medi della riga sovrastante e di quella sottostante o restituire il valore -1 se tale riga non esiste (non è possibile usare array ausiliari);

[V]*Es4. Data una matrice quadrata di stringhe, restituire una stringa ottenuta concatenando tutti gli ultimi caratteri delle stringhe presenti nella diagonale principale;

[V]*Es5. Stesso dell'Es4, però prendere la diagonale secondaria;

[V]*Es6. Data una matrice quadrata (es. 5x5) contenente bool, vi sarà un coniglio inizialmente al centro di questa matrice, che si sposterà di cella in cella, e lì dove ci sarà una carota (bool true), la mangia, trasformando il bool da true in false. Il coniglio parte da una vita pari a 100, e facendo salti casuali nella 4 direzioni si muove nella mappa, perdendo 1 di vita ogni salto, ed acquistando 10 di vita per ogni carota mangiata. Nel bordo della matrice, possiamo decidere se porre delle limitazioni, non facendo fuoriuscire il coniglio dalla matrice, oppure possiamo porre dei "teletrasporti" facendo ricomparire il coniglio dalla parte opposta della matrice (es. esce di sotto e rispunta nella parte superiore della matrice). Contare i salti effettuati dal coniglio, fino alla sua morte.

[V]*Es7. Data una matrice di float, dire se esistono due colonne le cui somme sono uguali, utilizzando un margine di errore e non utilizzando array ausiliari;

[V]*Es8. Generare una stringa casuale di lunghezza $3 < n < 8$, in cui ogni carattere è estratto tra (a,b,c,d)

-a con probabilità 1/7;

-b 3/7;

-c 2/7;

-d 1/7;

[V]*Es9. Data una matrice A di float, determinare un bool che indica se esistono in A 3 elementi, nella stessa colonna, ma in posizioni distinte, la cui somma è nulla;

[V]*Es10. Due stringhe si dicono simili se esse coincidono in almeno P posizioni, dove P è la metà (troncata) della lunghezza della stringa più

corta. Scrivere un metodo che prende in input una matrice di stringhe e restituisca un boolean che indichi se esiste una colonna della matrice

in cui sono presenti almeno 2 stringhe simili.

[V]*Es11. Scrivere un metodo che prenda in input una matrice A e un'array B, entrambe di stringhe, e restituisca un boolean che indichi se esiste in A una coppia di stringhe la cui concatenazione, privata delle occorrenze del carattere 'a', sia presente in B.

[V]*Es12. Generare una matrice NxN in cui il generico elemento è $(i+5)^{(j+2)}$. Senza utilizzare la funzione pow().

[]*Es13. Data una matrice di stringhe, dire se esiste una riga contenente due stringhe congruenti di ordine 3. Due stringhe si dicono congruenti di ordine N se il prefisso di lunghezza N della prima, \tilde{A} , è uguale al suffisso di lunghezza N della seconda;

[]*Es14. Dire se una stringa \tilde{A} è palindroma;

[]*Es15. Data una matrice quadrata di interi, dire se esiste una diagonale parallela alla principale, a somma nulla (incluendo la principale);

[V]*EsBonus (vecchio compito). Scrivere un metodo che prenda in input una matrice S di stringhe, e restituisca una copia S' di S in cui ogni stringa di lunghezza x:

- resta identica, se $x=8$;
- subisce l'inserimento a sinistra di $(8-x)$ caratteri '*', se $x<8$;
- subisce l'eliminazione dei primi $(x-8)$ caratteri, se $x>8$;