

Esercizi

Esercizio

- Si implementi una funzione che riceve in input una matrice $N \times M$ di float. Definito “picco” un numero circondato in tutte le posizioni intorno solo da numeri strettamente inferiori alla sua metà, la funzione conta il numero di “picchi” della matrice (attenzione a gestire correttamente gli elementi ai bordi della matrice).

Esercizio

- Si implementi una funzione che riceve in input una matrice $N \times M$ di float. Definito “pozzo” un numero circondato in tutte le otto posizioni intorno solo da numeri strettamente superiori al suo doppio, la funzione conta il numero di “pozzi” della matrice (attenzione a gestire correttamente gli elementi ai bordi della matrice, anche per loro bisogna controllare se sono pozzi).

Esercizio

- Definito il tipo punto (x,y,Nmonete)
scrivere una funzione che riceve in input una matrice M di punti di dimensione N*N e due coordinate x, y.
- La funzione deve navigare la matrice, partendo dal punto indicato dalle coordinate x e y e spostarsi ogni volta nelle coordinate indicate dal punto raggiunto. Nello spostarsi la funzione deve sommare tutti i valori di NMonete incontrati. La funzione procede allo stesso modo finché o non torna su un punto già toccato o trova coordinate non valide. A quel punto restituisce la somma delle monete raccolte.

Esercizio

- Si completi il programma sottostante, dove puzzle è una matrice di caratteri di R righe e C colonne che rappresenta un cruciverba. Le caselle nere sono rappresentate dal carattere '*'
- Si vogliono numerare le parole di almeno due lettere secondo la consueta numerazione dei cruciverba, partendo dalle celle in alto a sinistra e procedendo in ordine lessicografico
- Ad esempio, la matrice dichiarata nel programma corrisponde allo schema mostrato in figura (il numero di ciascuna parola è indicato nella cella contenente la prima lettera della parola)

¹ A	² B	I	*	³ I
⁴ T	O	*	⁵ E	R
⁶ O	B	A	M	A
S	*	*	⁷ I	Q

Il programma deve determinare la numerazione delle parole e stampare, per ogni numero, le coordinate (riga e colonna) della casella a cui si riferisce e se si tratta di una parola orizzontale, verticale o sia orizzontale che verticale.

Con la matrice dichiarata nel programma, l'output è come segue.

Parola 1: riga 0, colonna 0 verticale orizzontale

Parola 2: riga 0, colonna 1 verticale

Parola 3: riga 0, colonna 4 verticale

Parola 4: riga 1, colonna 0 orizzontale

Parola 5: riga 1, colonna 3 verticale orizzontale

Parola 6: riga 2, colonna 0 orizzontale

Parola 7: riga 3, colonna 3 orizzontale

Per l'implementazione si realizzi e si faccia uso di una funzione che controlli se alla riga r e colonna c inizi una parola di almeno due lettere nella matrice m lungo la direzione d (orizzontale se d è 0, verticale altrimenti). Si noti che una parola inizia in una certa casella se è preceduta, lungo la direzione considerata, o da una casella nera o dal bordo del puzzle.

Esercizio

- Definiamo *sequenza monotona crescente* in un vettore una sequenza di elementi contigui in cui ogni elemento in posizione $i+1$ è più grande di quello in posizione i .
4 5 7 è una sequenza monotona crescente di lunghezza 3
6 8 è una sequenza monotona crescente di lunghezza 2
- Definiamo *sequenza monotona decrescente* in un vettore una sequenza di elementi contigui in cui ogni elemento in posizione $i+1$ è più piccolo di quello in posizione i .
7 3 1 è una sequenza monotona decrescente di lunghezza 3
4 2 è una sequenza monotona decrescente di lunghezza 2
- Una lista di interi si dice ***uniformemente oscillante*** se tutte le ***sequenze monotone*** (crescenti o decrescenti) ***massime*** (cioè non contenute in altre sequenze monotone) che contiene hanno la stessa lunghezza. Codificare una funzione che riceve in ingresso una lista e restituisce True se è *uniformemente oscillante*, False altrimenti.
- Esempi:
4 5 7 3 1 5 9 4 3 è *uniformemente oscillante* (tutte le sequenze crescenti o decrescenti sono lunghe 3)
0 1 0 -1 0 1 0 -1 non è *uniformemente oscillante* (la prima sequenza 0 1 è più corta delle altre)

Esercizio

- Implementare una funzione per il lancio di dadi.
- La funzione prenda in ingresso il numero di facce del dado, e il numero di lanci che si vuole effettuare. La funzione simula i lanci del dado e stampa a video quante volte è uscita ciascuna faccia del dado. Si faccia inoltre in modo che la stampa risulti ordinata per numero di volte che è uscita una faccia in senso crescente.
- Ad esempio, dopo 5 lanci con un dado a 6 facce in cui sono usciti i numeri (1, 1, 5, 4, 6), si stampi a video:
La faccia 4 è uscita 1 volta
La faccia 5 è uscita 1 volta
La faccia 6 è uscita 1 volte
La faccia 1 è uscita 2 volte
- Attenzione, devono essere riportati sia il numero della faccia che il numero di volte che è uscita.
- Si usi la funzione

```
import random  
x=random.randint(1, sup);
```


Esercizio

- Definiamo *doppie* le parole in cui la prima metà è identica alla seconda (esempi: CECE, CANCAN, COUSCOUS) e *assonanti* le parole che sarebbero doppie se non avessero l'ultima lettera (esempi: AMAMI, MARMARA, PORPORA)
- Si definisca una funzione ... **doppia(...)** che riceve come parametro una stringa (che supponiamo valida e ben formata) e restituisce **True** se la stringa rappresenta una parola doppia, **False** altrimenti
- Si definisca una funzione ... **assonante(...)** che riceve una stringa e restituisce **True** se la stringa rappresenta una parola assonante, **False** altrimenti. **Si apprezzano, in particolare, soluzioni che riusano efficacemente la funzione definita al punto precedente**

Esercizio

- Si definisca una funzione che riceve in input un array a di 10 interi e una matrice m di 32×32 interi. L'array contiene una serie di interi tutti a 1 o a 0.
- La funzione calcola il numero decimale corrispondente alla codifica binaria rappresentata dalla sequenza di 1 e 0 dei primi 5 elementi dell'array e pone questo numero in una variabile intera x . Quindi calcola il numero decimale corrispondente alla codifica binaria rappresentata dalla sequenza di 1 e 0 dei rimanenti 5 elementi dell'array e pone questo numero in una variabile y .
- Infine la funzione considera gli elementi della riga con indice x in m e controlla se tra questi quelli maggiori di $m[x][y]$ siano di più di quelli minori di $m[x][y]$. Se questo è il caso, il programma restituisce 1, altrimenti 0.

Esercizio

- La matrice $M[6][156]$ contiene le 156 estrazioni del 1975 dei 6 numeri del SuperEnalotto.
- Scrivere una funzione che riceve la matrice M e restituisce quanti numeri tra 1 e 90 non sono mai usciti nel 1975.

Esercizio

- Si progetti e codifichi una funzione che riceve in ingresso un array a di interi non negativi.
- La funzione calcola il numero degli elementi dell'array che godono della seguente proprietà: il valore dell'elemento è pari al numero di elementi con valore inferiore considerando solo quelli in posizione precedente.
- Ad esempio, si consideri il seguente vettore:
- 1 2 1 **3** 6 **4** 14
- In questo caso solo i due numeri in grassetto (il 3 e il 4) soddisfano la proprietà descritta. Quindi la funzione restituisce 2.

Esercizio

- Si definisca una funzione che riceve in input due matrici **m1** e **m2** di $N \times N$ interi. La funzione calcola e restituisce il numero di elementi di **m2** che sono pari alla somma di due qualsiasi elementi di **m1** (se un dato elemento di **m2** è somma di più coppie di elementi di **m1**, viene contato comunque una volta sola)

Esercizio

- Si progetti e codifichi una funzione che riceve in ingresso un array **a** di interi. La funzione restituisce True se esiste almeno un elemento in **a** pari alla somma degli elementi che lo seguono diminuita della somma degli elementi che lo precedono. Altrimenti restituisce False.
- Ad esempio, nel vettore 1 2 1 **20** -6 16 14 il 20 soddisfa la proprietà descritta. Quindi la funzione restituisce True.
- Se invece si considera il vettore 1 2 1 20 -14 20 14 nessun numero soddisfa la proprietà descritta. Quindi la funzione restituisce False.

Esercizio

- Si scriva una funzione che riceve in input due array di dimensione N
- Si definisce *equilatero* un elemento di un vettore preceduto da tanti numeri pari più grandi quanti sono gli elementi dispari più piccoli che lo seguono. La funzione f deve copiare tutti gli elementi *equilateri* di a in b in posizioni contigue partendo dalla prima posizione di b senza lasciare buchi. Le posizioni finali di b che restano libere devono essere riempite di zeri.

Esercizio

- Implementare una funzione che riceve in input una matrice di matrici (una `matmat`) e un intero `k` e restituisce `1` se tra le matrici contenute nella `matmat` ricevuta ce ne sono almeno `k` che contengono il numero `k` almeno `k` volte
- Implementare una funzione `C` che riceve in input una matrice di matrici (una `matmat`) e restituisce il massimo numero `k` per cui è vero che tra le matrici contenute nella `matmat` ricevuta ce ne sono almeno `k` che contengono il numero `k` almeno `k` volte

Esercizio

- Si scriva una funzione che riceve in ingresso un vettore di numeri interi. Esso contiene dei valori sempre positivi o nulli (mai negativi). Una posizione contenente il numero 0 è da considerare vuota, cioè non contenente alcun valore.
- La funzione deve calcolare e stampare a video la media dei valori e poi deve modificare i valori che distano più di del 10% del valore della media dalla media stessa mettendoli a 0. La procedura deve essere ripetuta fino a che non ci sono più valori che vengono modificati.

Esercizio (tde 12-9-2011)

- La matrice di interi $M[N][N]$ contiene degli interi generati a caso.
- Scrivere una funzione che riceve la matrice M e restituisce quanti valori distinti contiene la matrice M

Esercizio

- Un semplicissimo (e fragile) sistema di cifratura dei messaggi consiste nel trasformare ogni lettera in una diversa particolare lettera. Il sistema che vi si propone agisce sulle sole lettere alfabetiche e dispone di un vettore chiave di interi, che definisce lo spiazzamento specifico da applicare alle 26 diverse lettere dell'alfabeto inglese. Per semplicità consideriamo solo le lettere maiuscole.
- Il primo elemento del vettore chiave indicherà lo spiazzamento da applicare alle a (se è 3, significa che tutte le a si tradurranno in d), il secondo lo spiazzamento per le b , ..., l'ultimo quello per le z (se è -13, si in m). Tutti gli altri eventuali caratteri (non alfabetici) del messaggio restano inalterati.

Si codifichi la funzione ...cipher(...) che riceve come parametri una chiave e un messaggio e alloca e restituisce il messaggio cifrato (che è una stringa dinamica).

Si dica qual è la minima modifica della funzione precedente con cui è possibile ottenere una funzione ...decipher(...) che effettui la trasformazione inversa.

Non tutte le chiavi sono altrettanto efficaci. Alcune possono essere ambigue e rendere più o meno indecifrabile il messaggio. Ad esempio una chiave $k = \{0, -1, -2, -3, \dots, -25\}$ è estremamente ambigua perché trasforma ogni lettera in una a . Si codifichi una funzione ...nonlossy(...) che controlla se una chiave permette sempre di ricostruire i messaggi non causando perdita d'informazione.