

Esercitazione di Fondamenti di Informatica del 11 Gennaio 2018

Esercizio 1.

Scrivere un programma che implementi la mescolata perfetta di un array con numero pari di elementi.

La mescolata perfetta consiste nel lasciare invariati i due estremi del mazzo e poi mettere la prima metà delle carte nelle posizioni pari del mazzo mescolato e la seconda metà delle carte nelle posizioni dispari del mazzo mescolato. Per esempio dovrà sostituire {11,22,33,44,55,66,77,88} con {11,55,22,66,33,77,44,88}.

Il programma dovrà leggere da input una sequenza pari di numeri interi e memorizzarla in un array di dimensione prefissata.

Modularizzare opportunamente il programma in funzioni, prevedendo un metodo per il mescolamento dell'array.

Questa funzione dovrà interfogliare la prima metà di un array con la seconda metà: ciò si ottiene usando un array temporaneo temp, e copiando alla fine temp nell'array originario.

Esempio: se l'array in input fosse:

11 22 33 44 55 66 77 88

l'output dovrà essere

11 55 22 66 33 77 44 88

Esercizio 2.

Scrivere un programma che legga da standard input una matrice M di dimensione n x n (con n costante definita) e determini se M ha un punto di sella. In caso affermativo, stampi le coordinate in cui si trova il punto di sella.

Un punto di sella è un elemento della matrice che è contemporaneamente il **minimo sulla sua riga e il massimo sulla sua colonna**.

Esempio: se la matrice in input fosse:

1 0 0 0
4 5 6 7
2 9 4 0
1 2 3 4

il programma dovrebbe stampare su standard output qualcosa di simile a:

Punto di sella in posizione (1,0).

Infatti l'elemento **4** in posizione (1,0) è il minimo della sua riga (la riga 1) e il massimo della sua colonna (la colonna zero)

Se la matrice in input fosse:

1 0 0 0
4 5 0 0
2 9 4 0
1 2 3 4

il programma dovrebbe stampare su standard output qualcosa di simile a:

Non esiste un punto di sella.

CAMPUS DI ARCAVACATA

Ponte Pietro Bucci Cubo 30B - 87036 Arcavacata di Rende (CS) tel. (+39) 0984 496400 fax (+39) 0984 496410

Esercizio 3.

Scrivere un programma, opportunamente modulato in funzioni, che, letta da input una sequenza di numeri interi (che possono essere positivi o negativi) terminata dallo 0 (zero), individui i punti di tipo A e B descritti di seguito.

Un numero costituisce un punto di tipo A se è dispari, e il numero che lo precede e quello che lo segue sono pari;

costituisce, invece, un punto di tipo B se è pari, e il numero che lo precede e quello che lo segue sono dispari.

Una volta individuati i punti di tipo A e B, il programma deve verificare se ogni punto di tipo A ha almeno un multiplo tra i punti di tipo B, e stampare su standard output la stringa "OK" in caso positivo; in caso negativo, si deve stampare "NO".

Esempio: se la sequenza in input fosse:

1 -4 11 3 70 5 -12 6 0

l'unico punto di tipo A è il numero 5 (infatti è dispari, ed è preceduto da 70 ed è seguito da -12, che sono entrambi pari);

i punti di tipo B sono invece -4 (preceduto da 1 e seguito da 5) e 70 (preceduto da 3 e seguito da 5).

L'unico punto di tipo A, che è 5, ha come multiplo 70 che sta tra i punti di tipo B: pertanto il programma, in questo caso, dovrebbe stampare "OK".

Esercizio 4.

Scrivere un programma opportunamente modularizzato in funzioni, che, letta una sequenza di numeri interi da input (lunga al massimo 100 elementi) terminata da un numero negativo, individui la sottosequenza **crescente** più lunga e la stampi su standard output, indicando anche quanto è lunga.

Esempio: data la sequenza

1 3 2 6 **1 4 5 6 9** 2 3 79 3 -1

i numeri in grassetto costituiscono la sottosequenza **crescente** di lunghezza massima; il programma dovrebbe stampare su standard output i numeri 1 4 5 6 9 e indicare che la lunghezza della sequenza è 5.