

Esercizio (inseriscielem): Dato un array *a* di riempimento *n* e cardinalità *NMAX* inserire un elemento *el* in posizione *pos*. Sia *a1* il nuovo array e *n1* il suo riempimento.

Esercizio (eliminaelem): Dato un array *a* di riempimento *n* e cardinalità *NMAX* eliminare l'elemento in posizione *pos*. Sia *a1* il nuovo array e *n1* il suo riempimento.

Esercizio (compatta): funzione *compatta()* che riceve un vettore *A* di taglia *MAXDIM* riempito con *n* elementi, e lo modifica in modo che elementi consecutivi uguali siano sostituiti da un'unica occorrenza. La funzione deve restituire il numero di elementi rimasti.

Esercizio (differenza): realizzare un programma che legga due array *A* e *B* (funzione *leggi*), rispettivamente di *n* ed *m* elementi, e stampi tutti gli elementi presenti in *B*, ma non in *A* (funzione *differenza*)

Esercizio (compatta2): scrivere un programma che legga *N* numeri interi da tastiera e li memorizzi in un vettore. Il numero *N* viene inserito dall'utente ed è minore di 20. Il programma deve generare un secondo vettore che compatta i numeri contenuti nel primo vettore. In particolare: ogni numero che compare ripetuto nel primo vettore deve comparire una sola volta nel secondo vettore; il programma deve visualizzare il contenuto del secondo vettore.

Ad esempio, si supponga *N* = 8 e si consideri la sequenza di numeri 1 18 3 0 24 3 6 0. Il programma deve visualizzare: 1 18 3 0 24 3 6.

Esercizio (matrice): scrivere un programma che legga una matrice *NxM*, *N* e *M* definiti dall'utente con *N* minore di *MAXROW* e *M* minore di *MAXCOL*, visualizzi la matrice, calcoli la somma, il minimo ed il massimo degli elementi della matrice (con 3 differenti funzioni).

Esercizio (intersezione): siano dati due vettori di interi inseriti da tastiera; la lunghezza dei due vettori è inserita dall'utente da tastiera. I due vettori possono avere lunghezze diverse, ma possono contenere al massimo 30 numeri. Si scriva un programma in C per generare un terzo vettore che contiene l'intersezione tra due vettori. Tale vettore deve contenere i numeri presenti in entrambi i vettori dati. Ad esempio, si assuma che siano stati inseriti i due vettori: 1 6 15 20 25, 2 20 18 6. Il programma deve visualizzare la sequenza 6 20.

Esercizio (sommamatrici): leggere due matrici e visualizzarne la somma; controllare che le dimensioni siano compatibili.

Esercizio (Armstrong): Un numero di Armstrong di tre cifre è un intero tale che la somma dei cubi delle sue cifre è uguale al numero stesso. Per esempio, $371 = 27 + 343 + 1$, quindi 371 è un numero di Armstrong. Scrivere una funzione che abbia come parametri input un array *A[]* di interi e il numero *n* di elementi presenti, azzeri i valori di *A[]* che non sono numeri di Armstrong, e infine stampi solo i valori rimanenti. Articolare la soluzione illustrando i passaggi seguiti: analisi e specifica, progettazione mediante raffinamento progressivo, traduzione in C.

Esercizio (filtrosom): scrivere una funzione *int filtrosum(int a[], int na, int b[], int nb, int c[], int nc);* che ha come parametri 3 array di interi *a[]*, *b[]*, e *c[]* con le rispettive lunghezze *na*, *nb*, *nc*. L'array *a[]* è la sorgente, *b[]* è il filtro, *c[]* è la destinazione. La funzione esamina l'array *a[]* e copia in *c[]* soltanto gli elementi per cui l'array filtro *b[]* contiene una coppia adiacente di elementi *b[k]* e *b[k+1]* che hanno la somma pari all'elemento da copiare. In altri termini, copia *a[i]* se esiste almeno un indice *k* per cui $b[k] + b[k+1] == a[i]$. La funzione restituisce il numero di elementi copiati. Si può assumere che $nc \leq na$, cioè che l'array destinazione sia abbastanza ampio da poterci copiare tutti gli elementi dell'array sorgente.

Esercizio (numeroperfetto): Fai inserire all'utente un numero. Attraverso una funzione verificare che il numero inserito sia perfetto (un numero è perfetto se equivale alla somma di tutti i suoi divisori. Es: $6=1+2+3$). SE è perfetto, calcolare il suo numero inverso (tipo, 135 diventa 531).

Esercizio (daesame): diciamo che un numero intero è *da esame* se la somma delle sue cifre decimali vale 6. Realizzare una funzione *int filtradaesame(int a[], int n)* che riceve come parametri un array *a[]* di *n* elementi interi e lo modifica azzerando tutti gli elementi che non sono numeri da

esame. La funzione restituisce il numero di elementi non nulli rimanenti, ossia la quantità di numeri da esame originariamente nell'array. Articolare la soluzione illustrando i passaggi seguiti: a. analisi e specifica, b. progettazione mediante raffinamento progressivo, c. codifica in c.

Tutorati

Esercizio (quadcubi): progettare e realizzare un programma che legge una sequenza di n interi e li memorizza in un array a , ne calcola il quadrato di ciascuno e lo memorizza in un array b , ne calcola il cubo, memorizzandolo in un array c , infine visualizza il contenuto di b e di c .

Esercizio (unisci): progettare e realizzare un programma che, dati due vettori di interi di dimensione N , ne costruisca un terzo di dimensione $2N$ i cui elementi di posizione pari siano gli elementi del primo vettore e gli elementi di posizione dispari siano gli elementi del secondo vettore.

Esercizio: progettare e realizzare un programma che, dato un vettore ($sorg[]$) di N double, ne generi un secondo ($dest[]$) così definito: l'elemento 0 di $dest$ è dato dal prodotto degli elementi 0 e 1 di $sorg$, l'elemento 1 di $dest$ è dato dal prodotto degli elementi 2 e 3 di $sorg$, e così via. Per esempio, se $sorg[] = \{0.5, 1.0, 1.5, 2.0\}$ si deve generare $dest[] = \{0.5, 3.0\}$. Si supponga che il numero di elementi di $sorg$ sia pari. Si visualizzi poi il vettore $dest$.