

ESERCIZIO 1

Definire una funzione ricorsiva che, dato un intero n , un carattere c e un array v di caratteri, controlli che il carattere c appaia esattamente n volte in v .

La funzione ricorsiva deve avere il seguente prototipo

```
int check (char *v1, int dim, int n, char c)
```

ESERCIZIO 2

Senza utilizzare costrutti iterativi, ma solo la ricorsione, definire una funzione che dato un array di interi ordinato calcola il numero di doppioni presenti nell'array.

Ad esempio, nel vettore rappresentato dalla seguente sequenza, i doppioni sono rappresentati in grassetto.

3 3 5 6 **6** **6** 9

il valore restituito quindi deve essere 3. Dato invece il vettore rappresentato dalla seguente sequenza

1 3 4 5 6 7

il valore restituito deve essere 0.

ESERCIZIO 3

Si definisca una funzione ricorsiva `ric-alf` che, dato un array `vet` di caratteri e la sua dimensione, restituisca il piu' grande carattere (nell'ordine alfabetico del C) che precede tutti i caratteri della sequenza. Se, ad esempio,

`vet`

`e l'array seguente

'w' 'j' 'd' 'l' 'c' 'q' 'q' 'e'

la funzione deve restituire il carattere

'b'.

N.B. la funzione

`ric-alf`

deve essere la funzione principale, non deve avere altri parametri e deve essere ricorsiva

ESERCIZIO 4

Si definisca in C una funzione ricorsiva che dato un array di interi, costituito da un numero dispari di elementi, calcola true se la somma degli elementi equidistanti dall'elemento mediano e uguale all'elemento mediano. Ad

esempio: se l'array contiene:

3 0 7 4 8 4 1 8 5

la funzione calcola true, essendo l'elemento mediano 8 (in posizione 4) e avendo le coppie (3,5), (0,8) (7,1) e (4,4)

tutte come somma 8.