

Fondamenti di Programmazione A

Prova di programmazione (C++)

*Evidenziare la strutturazione del programma mediante indentazione (OBBLIGATORIO)
Inserire anche adeguati commenti*

Sia *S* il tipo di una struttura dati (`struct`) costituita da due campi, *c* di tipo stringa (lunghezza max. 100) e *r* di tipo numero reale.

(a) Realizzare una funzione `void` di nome `compatta` che presi come suoi parametri un array *A* di elementi di tipo *S* con la sua lunghezza *n*, una stringa *Aux* di *k* caratteri '0' o '1' ($k \geq n$), e un array *B* di elementi di tipo *S* con la sua lunghezza *m*, copia gli elementi di *A* in *B* nel modo seguente: l'*i*-esimo elemento di *A* viene copiato in *B* se e solo se l'*i*-esimo carattere di *Aux* è '1'. La funzione restituisce in *m* il numero di elementi memorizzati in *B*.

N.B. La funzione deve essere obbligatoriamente dichiarata `void`. Gli eventuali elementi di *Aux* di indice $\geq n$ vengono ignorati. Si assuma che la condizione $k \geq n$ sia garantita dal programma chiamante (dunque, non deve essere controllata dalla funzione).

(b) Realizzare un programma principale di prova che dichiara una stringa *C* di 30 caratteri inizializzata con il valore "101011000101000111111000111010" e quindi legge da std input una sequenza di coppie di elementi di tipo *S*, fino a leggere la coppia \$\$\$ 0, e le memorizza in un array *V1* di elementi di tipo *S* (di dimensione massima 30). Il programma quindi richiama la funzione `compatta` con *V1* e *C* che corrispondono, rispettivamente, ad *A* e *Aux*, provvedendo quindi a stampare su std output l'array compattato (N.B. stampare solo gli elementi significativi ...).

Utilizzare soltanto stringhe "tipo C".

Esempio.

Sequenza di input:

```
aa 1.5
b 2.4
ccc 3.7
dd 5
$$$ 0
```

Array compattato:

```
aa 1.5
ccc 3.7
```