

**ESERCIZI
DI PROGRAMMAZIONE
DA SVOLGERE
INDIVIDUALMENTE
- parte 2 -**

1: ESERCIZI DI BASE

- di livello di difficoltà inferiore rispetto all'esame -
- propedeutici per affrontare esercizi più complicati -

- B1) Scrivere un programma che riceva in ingresso due vettori vet_1 e vet_2, ciascuno di 10 elementi interi e produca in uscita un vettore vet_3 in cui ogni elemento sia il minimo degli elementi di ugual posizione in vet_1 e vet_2.
- B2) Scrivere un programma che riceva in ingresso al massimo 10 interi e li stampi in ordine inverso (nota: simile all'esercizio 1 dei lucidi, ma in questo caso si può fare a meno di definire un vettore vet_2)
- B3) Scrivere un programma che acquisisca da tastiera al massimo 50 numeri interi, interrompendo l'acquisizione quando venga immesso il numero 0. Salvare i numeri acquisiti in un vettore e, successivamente, stampare il minimo, il massimo e la media.
- B4) Scrivere un programma che acquisisca da tastiera un vettore di 20 interi compresi tra 5 e 10, quindi stampi per ogni elemento quante volte compare nel vettore.
- B5) Sviluppare un programma che acquisisca dall'utente una stringa e la salvi in un vettore di caratteri. Successivamente, generi e stampi un altro vettore che contiene la stringa acquisita dalla quale siano eliminate tutte le vocali.

2: ESERCIZI DA ESAME

Esercizio 1

Esercizio 9 del Tema d'esame del 16 giugno 2009

Esercizio 2

Esercizio 8 del Tema d'esame del 14 gennaio 2010

Esercizio 3

Si sviluppi un programma in linguaggio C che acquisisca in ingresso due numeri binari di 8 bit e li memorizzi in due vettori *num1* e *num2*.

Successivamente ne faccia la somma algebrica interpretandoli come numeri rappresentati in complemento a due, ponendo il risultato (rappresentato in complemento a due) in un vettore *somma* e segnalando all'utente se c'è stato overflow.

Esercizio 4

Esercizio 8 del Tema d'esame del 2 luglio 2009

Esercizio 5

Si sviluppi un programma in linguaggio C che, acquisite due stringhe

```
char caratteri[5];
```

```
char stringa[25];
```

elimini dalla seconda stringa tutti i caratteri presenti nella prima.

Se una stringa occupa meno posizioni di quelle disponibili nel vettore (rispettivamente 5 e 25), si suppone che sia allineata a sinistra e che tutti i rimanenti posti liberi a destra siano occupati dal carattere zero.

Ad esempio, se `caratteri="pabab"` e `stringa="pasta all'amatriciana0000"` il programma deve modificare `stringa` in modo che risulti `"st ll'mtricin00000000000000"`.

SOLUZIONI DISPONIBILI

Gli studenti sono fortemente incoraggiati a sperimentare le proprie soluzioni al calcolatore. Nel correggere gli errori che inevitabilmente saranno commessi, si suggerisce di non procedere per tentativi bensì di identificare precisamente la causa del comportamento inatteso del programma sviluppato.

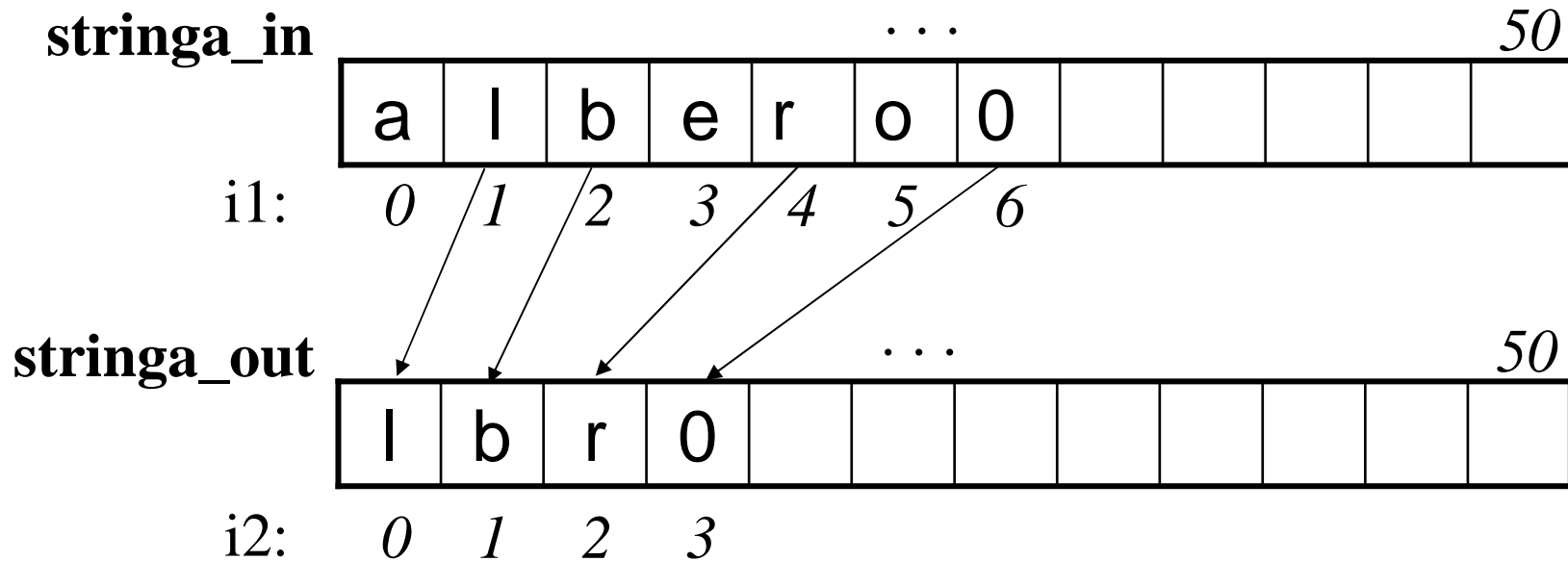
Esercizio B5

Sviluppare un programma che acquisisca dall'utente una stringa e la salvi in un vettore di caratteri. Successivamente, generi e stampi un altro vettore che contiene la stringa acquisita dalla quale siano eliminate tutte le vocali.

Esercizio B5

Sviluppare un programma che acquisisca dall'utente una stringa e la salvi in un vettore di caratteri. Successivamente, generi e stampi un altro vettore che contiene la stringa acquisita dalla quale siano eliminate tutte le vocali.

Algoritmo



➡ Ho bisogno di due indici: $i1$ e $i2$

ACQUISISCI stringain

```
i1=0;                                //contatore su stringa di ingresso
i2=0;                                //contatore su stringa di uscita
while(stringain[i1]!='\0'){
    Se è una consonante, inserisci il carattere stringain[i1] in stringaout
    i1++;
}
```

//quando esce devo ancora inserire 0 in stringaout

Inserisci 0 in stringaout

STAMPA stringaout

```

int i1, i2, i;
char stringain[50], stringaout[50];

printf("Inserisci stringa\n");
scanf("%s", stringain);

i1=0;                                     //contatore su stringa di ingresso
i2=0;                                     //contatore su stringa di uscita
while(stringain[i1]!='\0'){
    if(stringain[i1]!='a' && stringain[i1]!='e' && stringain[i1]!='i' && ...)
        stringaout[i2++]=stringain[i1];
    i1++;
}                                         //quando esce i2 contiene il posto in cui andrebbe inserito 0

stringaout[i2]='\0';
printf("Stringa di uscita:  %s\n", stringaout);

```