

INFORMAZIONI IMPORTANTI:

I programmi richiesti devono leggere l'input da un file **input1**, **input2**, ecc. e devono scrivere l'output su un file output1, output2, ecc. I valori sui file di input e di output sono sempre separati da uno spazio. In ogni esercizio sarà descritto in dettaglio cosa vada assunto sul contenuto del file di input e cosa vada scritto su quello di output. Il file che contiene la soluzione dell'esercizio (1) **deve chiamarsi esercizio1.cpp e quello del (2) deve chiamarsi esercizio2.cpp**. Per ogni esercizio trovate nella vostra home un file di input ed un corrispondente file di output. Si tratta solo di esempi. Dovreste provare i vostri programmi anche con altri file di input. I vostri programmi dovrebbero compilare e per gli input dati dovrebbero produrre output uguali a quelli dati.

ESERCIZI:

Si richiede di scrivere (come commenti dopo i programmi) pre- e postcondizioni per ogni programma e invarianti per ogni ciclo. Si chiede anche di delineare la dimostrazione di correttezza del ciclo principale dell'esercizio (1).

1) Si tratta di scrivere un programma che dichiari `char A[5][10]` e che poi legga da `input1` un intero `dim` ( $0 < \text{dim} \leq 50$ ) e dopo legga `dim` caratteri che andranno inseriti in `A` riempiendo le righe in ordine, prima la riga 0, poi la 1 e così via fino ad avere inserito tutti i `dim` caratteri. Si osservi che in generale `A` sarà solo parzialmente riempita e che l'ultima riga che contiene qualche elemento (potrebbe anche essere l'unica riga presente) avrà in generale meno di 10 elementi. Dopo la fase di input, il programma dovrà determinare se in `A` esiste qualche riga che contiene un numero pari di 'a'. Se una tale riga esiste, allora il programma dovrà scrivere su `output1` il valore `true` seguito dall'indice minimo di una tale riga. Se `A` non ha alcuna riga con questa proprietà, allora il programma deve scrivere `false` su `output1`. Il programma deve considerare solo gli elementi definiti di `A` (cioè i primi `dim` elementi).

**Esempio:** Supponiamo per semplicità che `A` sia `char [3][6]`, che `dim = 16` e che dopo aver letto i 16 caratteri `A` sia come segue `A = [['a','c','b','a','c','a'] ['a','b','b','v','z','a'] ['a','a','a','a']]`. Per un tale array il programma deve scrivere su `output1`, `true` e 1 visto che la riga di indice 1 è la prima che contiene un numero pari di 'a' (2).

2) Il programma richiesto deve eseguire lo stesso input del precedente (ma dal file `input2`), ma in caso ci siano in `A` delle righe che contengono un numero pari di 'a', deve scrivere su `output2` l'indice massimo di una tale riga.

**Esempio:** Sia `A` come nell'esempio precedente, allora il programma deve scrivere `true` e poi 2 su `output2`, visto che 2 è l'indice massimo delle righe che contengono un numero pari di 'a'. Si osservi che la riga 2 non è completa, ma che la parte definita contiene 4 'a'.