

esercizio 31/1 (1)

LETTURA

```
ifstream INP("input1");  
ofstream OUT("outpu1");  
char A[5][10], *p=*A;  
int dim;  
INP>>dim;  
cout<<dim<<endl;  
for(int i=0; i<dim; i++)  
{INP>>p[i]; cout<<p[i]<<' ';}  
cout<<endl;
```

```
int nr=dim/10, nef=dim%10;
```

RIGHE PIENE

```
bool trovata=false;
```

```
for(int i=0; i<nr && !trovata; i++) //R1
```

```
{
```

```
    int c=0;
```

```
    for(int j=0; j<10; j++) //R2
```

```
        if(A[i][j]=='a')
```

```
            c++;
```

```
    // POST2
```

```
    if(!(c%2))
```

```
        {OUT<< true << ' ' << i << endl; trovata=true;}
```

```
} //POST1
```

POST1=(trovata sse esiste riga [0..nr-1] con N. pari di 'a')&&(trovata=> OUT contiene true e indice min di tale riga)

R1=(0<=i<=nr)&&(le righe [0..i-2] non hanno N. pari di 'a')&&(la riga i-1 contiene N. pari di 'a' <==> trovata)&&(trovata => OUT contiene true e i-1)

R2=(0<=j<=10)&&(c= N. di 'a' in A[i][0..j-1])
POST2=(c=N.di 'a' in A[i])

prova ciclo interno

1. Condizione iniziale: $c=0$; $j=0$, con questi valori $R2 = (0 \leq c \leq 10) \ \&\& \ (0 \leq N. \text{ di 'a' in } A[i][0..-1])$ essendo $[0..-1]$, vuoto il n. di 'a' che vi compare è 0. Quindi vale $R2$ la prima volta.

2. Invarianza: $R2 \ \&\& \ (j < 10) \rightarrow \text{if}(A[i][j] == 'a') \ c++; j++ \rightarrow R2$
dato che $j < 10$ $A[i][j]$ è un elemento della riga i , dopo averlo confrontato con 'a', se è uguale si esegue $c++$. Visto che $R2$ era vero prima di farlo, c era giusto per $A[i][0..j-1]$ e quindi dopo il condiz, c è giusto per $A[i][0..j]$, quindi dopo $j++$, c sarà giusto per $A[i][0..j-1]$, come in $R2$ e inoltre è vero che $j \leq 10$. Quindi vale $R2$ come volevamo.

3. condizione d'uscita: $R2 \ \&\& \ !(j < 10) = (c = n. \text{ di 'a' in } A[i][0..j-1]) \ \&\& \ (j = 10) \Rightarrow (c = n. \text{ di 'a' in } A[i][0..9])$

Prova del ciclo esterno:

1. condizione iniziale: $i=0$ e $trovata=false$ e quindi $R1$ è vera: $(0 \leq 0 \leq nr)$. Le righe $[0..0-2]$ non ci sono e neanche la riga $[0-1]$ esiste e non può contenere 'a' e quindi è giusto che $trovata = false$. Visto che $trovata=false$ l'ultima condizione di $R1$ è trivialmente verificata.

2. invarianza: assumiamo $R1 \ \&\& \ (i < nr \ \&\& \ !trovata)$. Essendo $trovata=false$, la riga $i-1$ non andava. $i < nr$ e quindi la riga i va considerata.

Eseguiamo il corpo 1 volta:

il corpo considera la riga i e con il ciclo interno calcola in n. di 'a' in questa riga. Vedi Post2. Il successivo condizionale non fa nulla nel caso la riga i -esima non contenga un n. pari di 'a', mentre altrimenti mette trovata = true ed esegue l'output (true i).

quindi dopo il condizionale vale:

$(0 \leq i < nr) \&\& (le\ righe\ [0..i-1]\ non\ hanno\ N.\ pari\ di\ 'a') \&\& (la\ riga\ i\ contiene\ N.\ pari\ di\ 'a' \iff trovata) \&\& (trovata \Rightarrow OUT\ contiene\ true\ e\ i)$

è R1 ma con i al posto di $i-1$. L'incremento $i++$ trasforma questa asserzione in R1

3. condizione d'uscita: $R1 \ \&\& \ !(i < nr \ \&\& \ !trovata) \Rightarrow POST1$
 $!(i < nr \ \&\& \ !trovata) = i \geq nr \ || \ trovata$ si considerano i 2 casi separatamente:

i) $trovata$ è vero, da $R1$ segue che per $0 \leq i \leq nr$

(le righe $[0..i-2]$ non hanno N. pari di 'a') $\&\&$ (la riga $i-1$ contiene N. pari di 'a') $\&\&$ (OUT contiene true e $i-1$)
quindi vale $POST1$

ii) $!trovata \ \&\& \ i \geq nr$ che assieme a $R1 \Rightarrow i = nr$. Da $R1$ si ha:

(le righe $[0..nr-2]$ non hanno N. pari di 'a') $\&\&$ (la riga $nr-1$ non ha N. pari di 'a')

\Rightarrow nessuna riga $[0..nr-1]$ di A con N. pari di 'a'
vale $POST1$

ULTIMA RIGA

```
if(!trovata)
{ if(nef>0)
  {
    int c=0;
    for(int i=0; i<nef; i++) //R2'
      if(A[nr][i]=='a')
        c++;
    //POST2'
    if(!(c%2))
      {OUT<< true << ' '<< nr<< endl; trovata=true;}
    } //POST3
  if( !trovata )
    OUT << false << endl;
}
```

R2' e POST2' sono uguali a R2 e POST2 con la sola differenza che 10 deve essere sostituito con nef

POST3 = (trovata <==> A[nr] ha N. pari di 'a')
&&(trovata=> OUT contiene true e nr)