

mercoledì 25 gennaio 2012

Tutto il materiale in:

www.math.unipd.it/~gilberto/programmazione

Consegnare

Si consegna **sempre un solo file** che si deve chiamare **"esercizio.cpp"**

Messo in una cartella che contiene solo quel file (e al più a.out)

Il comando di consegna è:

consegna esame

Chi non consegna niente **viene tolto dalla lista degli iscritti** al corso

Potrà comunque riiscriversi se pensa di venire alle prossime esercitazioni

Se si consegnano 2 o più esercizi nello stesso file ci potrebbero essere diversi main.

Per compilare mettere tutti i main meno 1 sotto commento

Con `//` per riga e `/*.....*/` per commenti che si possono estendere a piacimento.

Esercizio 1: dato un array
char B[5][10] determinare l'indice
minimo di una **riga** di B che
contiene un numero dispari di 'a'. B
non va modificata.

Esercizio 1:

PRE=(B[5][10] definito, $B=\underline{B}$,
SI(B[0..4]))

POST=($0 \leq \text{index} \leq 4$ è l'indice
minimo tale che B[index] contiene
un numero dispari di 'a', $B=\underline{B}$;

SI(B[i..j]) = c'è una riga tra i..j con
n. dispari di 'a', quando $i > j$ allora
false

```
R=(guardate le righe 0..i-1, SI(B[0..i-2])=false,  
ok => (SI(B[i-1]) && indice=i-1),  
!ok => SI(B[i..4]))  
bool ok=false; int indice;
```

```
for(int i=0; !ok; i++)  
{int conta=0;
```

```
    for(int j=0; j<10; j++) // conta=n. 'a'  
        if(B[i][j]=='a') conta++; // in B[i][0..j-1]
```

```
    if(conta%2)  
        {ok=true; indice=i;}  
}
```

Ragionamento:

$!ok \Rightarrow SI(B[i..4])$ implica che ok può restare false solo per $i \leq 4$, perché se ok fosse false con $i \geq 5$, allora dovrebbe essere vero $SI(B[5..4])$ che è falso per definizione.

Quindi certamente ok diventa vero con $i \leq 5$ e quindi $index$ sarà tra 0 e 4, come richiesto dalla POST

Inoltre da $SI(B[0..i-2]) = false$ e $ok \Rightarrow (SI(B[i-1]) \ \&\& \ index = i-1)$, e da $ok = true$ (negazione test di permanenza) segue che $index$ è il minimo indice di riga che soddisfa la condizione, come richiesto dalla POST.

Esercizio 2: data `char B[5][10]`
calcolare l'indice minimo di una
colonna di B che contiene lo stesso
numero di `'a'` e di `'b'`. B non va
modificato.

Scrivere PRE e POST, invarianti dei
2 cicli e main

Non assumiamo che B abbia
colonna OK

PRE=(B[5][10] definita, B=B)

POST=(B=B,
SI(B[][0..9]) => indice è il minimo in 0..9
t.c. SI(B[][indice])
!SI(B[][0..9]) => indice=-1
)

$R = (0 \leq i \leq 10, \text{esamine le colonne } 0..i-1,$
 $SI(B[][0..i-2]) = \text{false}$
 $ok \Leftrightarrow ((SI(B[][i-1]) \ \&\& \text{indice} = i-1),$
 $!ok \Rightarrow \text{indice} = -1)$

```
int indice=-1; bool ok=false;
for(int i=0; i<10 && !ok ; i++)
{
```

che fare qui ? Ciclo sulla colonna i-esima
for(int j=0; j<5;j++) che conta la differenza tra 'a'
e 'b'

$R1 = (\text{diff} = \text{differenza tra n. di 'a' e 'b' in } B[0..j-1][i])$

```
}
```

Corpo del ciclo con
indice i che scorre le
colonne

```
int diff =0;
for(int j=0; j<5;j++) //R1
{
    if(B[j][i]=='a')
        diff++;
    else
        if(B[j][i]=='b')
            diff--;
}

if(!diff)
{ok=true; indice=i;}
```

```
int indice=-1; bool ok=false;
for(int i=0; i<10 && !ok; i++)
{
    int diff =0;
    for(int j=0; j<5;j++) //R1
    {
        if(B[j][i]=='a')
            diff++;
        else
            if(B[j][i]=='b')
                diff--;
    }

    if(!diff)
    {ok=true; indice=i;}
}
```

Verifica della condizione d'uscita:

$R \ \&\& \ !(i < 10 \ \&\& \ !ok) = R \ \&\& \ (i = 10 \ || \ ok)$

- a) Se ok allora R garantisce che index è la minima colonna che soddisfa la condizione, come richiede la POST.
- b) Se $i = 10 \ \&\& \ !ok \Rightarrow R$ implica che $index = -1$ e che $!SI(B[][0..9])$, come richiesto dalla POST è verificata