

soluzioni 23 e 24 gennaio

```
int inizio=0; bool trovato=false;
while (inizio<dim-dimP+1 && !trovato) //R1
{ int k=0;
  while(k<dimP && T[inizio+k]==P[k])//R2
    {k++;}
  //POST2
  if(k==dimP)
    {trovato=true;}
  else
    inizio++;}
//POST1
```

$R1 = (0 \leq \text{inizio} \leq \text{dim} - \text{dim}P + 1) \ \&\&$
 $(\text{trovato} \Rightarrow \text{inizio} < \text{dim} - \text{dim}P + 1 \ \&\& \text{ match inizia in } T[\text{inizio}] \ \&\& \text{ è l'unico match che inizia in } T[0..\text{inizio}]) \ \&\&$
 $(\neg \text{trovato} \Rightarrow \text{nessun match inizia in } T[0..\text{inizio}-1])$

$R2 = (0 \leq k \leq \text{dim}P) \ \&\& (T[\text{inizio}..\text{inizio}+k-1] = P[0..k-1])$

$\text{POST2} = (k = \text{dim}P \Rightarrow \text{match che inizia in inizio}) \ \&\& (k < \text{dim}P \Rightarrow T[\text{inizio}+k] \neq P[k])$

$\text{POST1} = (\text{trovato} \Rightarrow \text{c'è match che inizia in } T[\text{inizio}] \text{ e niente con inizio in } [0..\text{inizio}-1]) \ \&\& (\neg \text{trovato} \Rightarrow \text{nessun match in } [0..\text{dim}-1])$

POST1=(trovato=> c'è match che inizia in inizio e niente con inizio in [0..inizio-1]) &&(!trovato => nessun match in [0..dim-1])

if(trovato)

OUT<< true <<' '<<inizio<<endl;

else

OUT<<false<<endl;

POST =(se esiste match e il primo è in inizio => OUT contiene true+inizio) &&(se non esiste match => OUT contiene false)

```
int k=0;
```

```
R2=(0<=k<=dimP) && (T[inizio..inizio+k-1]=P[0..k-1])
```

```
while(k<dimP && T[i+k]==P[k])  
  {k++;}
```

```
POST2=(k=dimP=> match che inizia in  
inizio)&&(k<dimP => T[inizio+k]!=P[k])
```

condizione iniziale, invarianza e terminazione

```
R1=(0<=inizio<=dim-dimP+1) &&  
(trovato=> inizio <dim-dimP+1 && match inizia in T[inizio] && è  
l'unico match che inizia in T[0..inizio]) &&  
(!trovato=> nessun match inizia in T[0..inizio-1]) &&(inizio<dim-  
dimP+1 && !trovato)
```

```
{ int k=0;  
while(k<dimP && T[inizio+k]==P[k])//R2  
  {k++;}
```

```
\\POST2=(k=dimP=> match che inizia in  
inizio)&&(k<dimP => T[inizio+k]!=P[k])
```

```
if(k==dimP)  
  {trovato=true;}  
else  
  inizio++;}
```

INVARIANZA

R1

Condizione d'uscita del ciclo 1

$R1 = (0 \leq \text{inizio} \leq \text{dim} - \text{dim}P + 1) \ \&\&$
 $(\text{trovato} \Rightarrow \text{inizio} < \text{dim} - \text{dim}P + 1 \ \&\& \text{match inizia in } T[\text{inizio}]$
 $\&\& \text{ è l'unico match che inizia in } T[0..\text{inizio}]) \ \&\&$
 $(\text{!trovato} \Rightarrow \text{nessun match inizia in } T[0..\text{inizio}-1])$

$\&\& \text{ ! (inizio} < \text{dim} - \text{dim}P + 1 \ \&\& \text{!trovato)}$

\Rightarrow

$\text{POST1} = (\text{trovato} \Rightarrow \text{c'è match che inizia in } T[\text{inizio}] \text{ e niente con}$
 $\text{inizio in } [0..\text{inizio}-1] \ \&\& (\text{!trovato} \Rightarrow \text{nessun match in } [0..\text{dim}-$
 $1])$

PER CASI: (a) trovato e (b) $\text{inizio} \geq \text{dim} - \text{dim}P + 1$

esercizio (2) del 24/1

calcolare la sottosequenza crescente max (SCM) in $A[0..dim-1]$

idea: con 2 cicli annidati si tiene traccia della sottoseq max fino a un certo punto ($ic=fc$, inizialmente 0) e col ciclo interno si determina la prossima sottoseq massima facendo crescere fc che terminerà con $fc=dim-1$ oppure con $A[fc] > A[fc+1]$

$R2=(ic \leq fc \leq dim-1) \&\& (A[ic..fc] \text{ è crescente})$

$while(fc < dim-1 \&\& A[fc] \leq A[fc+1]) fc++;$

$POST2=(A[ic..fc] \text{ è crescente}) \&\& (fc=dim-1 \mid \mid A[fc] > A[fc+1])$


```

int ib=0, fb=0, ic=0, fc=0, lb=1, lc;
while(fc < dim-1) //R1=(A[ib..fb] SCM in A[0..fc]) &&(A[ic..fc]
SC)
{
  while(fc<dim-1 && A[fc]<=A[fc+1]) fc++;
  POST2=(A[ic..fc] è crescente) &&(fc=dim-1 || A[fc]>A[fc+1])

  lc=fc-ic+1;
  if(lc > lb)
    {ib=ic; fb=fc; lb=lc;}

  if(fc < dim-1)
    ic=fc=fc+1;
} //POST1=(A[ib..fb] SCM in A[0..dim-1])

```