

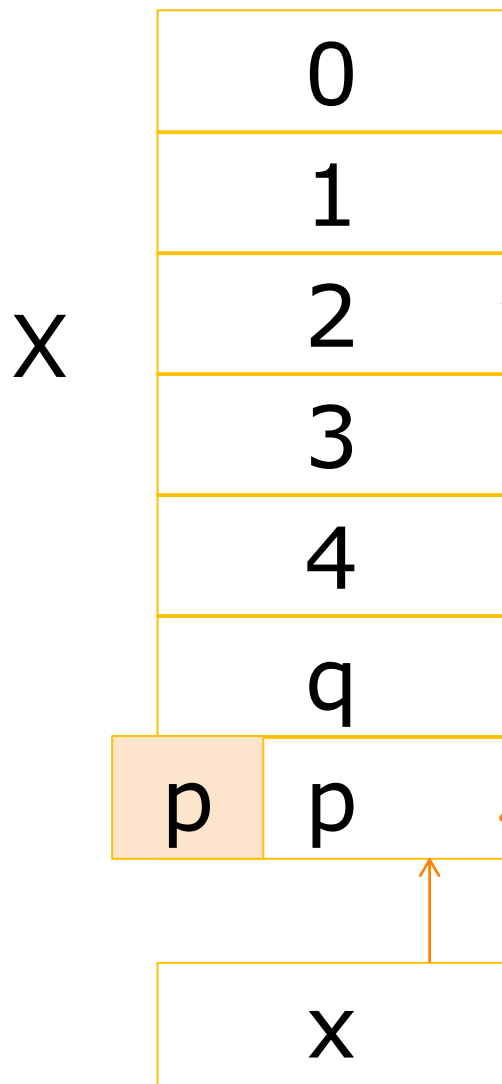
mercoledì 29 febbraio 2012

II COMPITINO

Teoria) Si consideri il seguente programma e si dica se è corretto o meno, spiegando il motivo della risposta:

```
int * & F(int** & p){int***x=&p;  
*x=p+1; return **x;}
```

```
main(){ int X[5]={0,1,2,3,4},  
*q=X+2, **p=&q; F(p)=q+2;  
cout<<*q<<**p<<endl;}
```



$*x = p + 1;$

p punta a se stesso o comunque ad un'area della RAM su cui non c'è controllo
return $**x$ restituisce proprio quest'area
e $F(p) = q + 2$
ci mette dentro

Programmazione) Dato un array `int A[lim]`, con $\text{lim} > 0$, un **intervallo** di `A` è specificato da una coppia (i, n) tale che $0 \leq i \leq i+n-1 < \text{lim}$ ed è costituito dagli elementi `A[i], A[i+1]...A[i+n-1]`. Il **bilancio** dell'intervallo (i, n) è la differenza tra il **numero** di valori positivi presenti in `A[i], A[i+1]...A[i+n-1]` ed il **numero** di quelli negativi.

Per esempio, se $A = [1, -2, -3, 2, -4, 3, 2, 4, -1]$ allora il bilancio dell'intervallo $(2, 3) = [-3, 2, -4]$ è -1 visto che contiene un valore positivo e 2 negativi.

Data la seguente dichiarazione: `struct interv{int inizio, lunghezza;};` si chiede di scrivere una funzione `interv F(int*A, int*B, int lim)` che soddisfi le seguenti pre- e post-condizioni:

PRE = (`A[0..lim-1]` e `B[0..lim-1]` sono definiti e $\text{lim} > 0$)

POST = (restituisce valore `Y` di tipo `interv` tale che `A[Y.inizio]...A[Y.inizio+Y.lunghezza-1]` è un intervallo di `A` che ha bilancio massimo).

NOTA: Dovete usare l' array B che va riempito in modo che valga la seguente condizione: per ogni j in $[0..lim-1]$, $B[j]$ = bilancio massimo degli intervalli di A che iniziano in posizione j .

Il valore da mettere in $B[lim-1]$ è immediato da calcolare. Usando $B[lim-1]$ è semplice calcolare $B[lim-2]$ e, nella stessa maniera, si possono calcolare $B[lim-3]$, $B[lim-4]$ fino a $B[0]$. Quando B è riempito, calcolare il valore $interv$ che rappresenta l'intervallo a bilancio massimo in A , è un gioco da ragazzi. La vostra soluzione dovrebbe usare solo gli array A e B . Cercate di organizzare il calcolo in operazioni più semplici e realizzatele con apposite funzioni.

Qualora ci fossero più intervalli di A con bilancio maggiore o uguale agli altri, F deve restituire uno qualsiasi di essi.

```
int traduci(int x)
{
    if(x >= 0)
        return 1;
    else
        return -1;
}
```

```
POST = ( x >= 0 ==> 1 ) && ( x < 0 ==> -1 )
```

//PRE=(B[i+1] è bilancio max intervalli con
inizio in i+1)

```
void fixB(int*A,int*B, int i)
{
    if(B[i+1]>0)
        B[i]=traduci(A[i])+B[i+1];
    else
        B[i]=traduci(A[i]);
}
```

//POST=(B[i] è bilancio max intervalli con
inizio in i)

```
interv  F(int*A,int*B,int lim)
{ B[lim-1]=traduci(A[lim-1]);
  int pos=lim-1;
  for(int i=lim-2; i>=0; i--) //R
  {
    fixB(A,B,i);
    if(B[i]>B[pos])
      pos=i;
  }
```

R=(per ogni a in $[i+1..lim-1]$, B[a] contiene
bilancio max interv con inizio in a)&& (B[pos] è
max in B[i+1..lim-1] && $(-1 \leq i \leq lim-2)$)

PRE=(A[0..lim-1] è def., $0 \leq \text{inizio} < \text{lim}$, bilancio ≥ -1)

int trovaL(int*A, int inizio, int bilancio, int lim)

```
{int somma=0, lung=0;  
  do  
  {  
    somma=somma+traduci(A[inizio]);  
    inizio++; lung++;  
  }  
  while(somma<bilancio && inizio<lim);  
  
  if(somma != bilancio) throw (1);  
  return lung;  
}
```

POST=

(restituisce lung tale che A[inizio],..., A[inizio+lung-1] ha bilancio bilancio)

```
interv F(int*A,int*B,int lim)
{ B[lim-1]=traduci(A[lim-1]);
  int pos=lim-1;
  for(int i=lim-2; i>=0; i--)
  {
    fixB(A,B,i);
    if(B[i]>B[pos])
      pos=i;
  }
  interv I; I.inizio=pos;
  I.lunghezza=trovaL(A,pos,B[pos],lim);
  return I; }
//A[I.inizio],...,A[I.inizio+I.lunghezza-1] è
a bilancio massimo
```