

correttezza dei programmi

un esercizio

regola del while e for

altri esercizi

problema:

leggere 10 interi da cin e calcolare il
minimo ed il massimo man mano che si
fanno le letture

PRE= (cin contiene almeno 10 interi)

POST=(min è il minimo dei 10 valori letti
e max è il massimo)

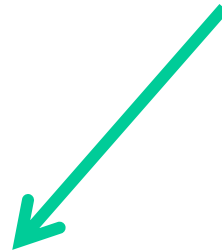
leggere 10 interi da cin è facile

avremo variabili max e min e una variabile
int x per leggere da cin e una i per contare
le letture

PRE

ciclo while che dovrebbe fare

POST



$R = (0 \leq i \leq 10, \text{ letti i valori da cin, min e max sono il minimo ed il massimo degli i valori})$

- una cosa da decidere è come inizializzare max e min

consideriamo 2 idee:

(i) leggiamo il primo valore in x e poi
min=max=x;

(ii) max=MIN_INT e min= MAX_INT

- per il calcolo di min e max viene naturale pensare che:

(iii) se il prossimo valore letto migliora max allora inutile guardare il min e viceversa

iniziamo con (i) e (iii)

```
main() //PRE
```

```
{
```

```
    int x, min, max, i=1;
```

```
    cin >> x; min=max=x;
```

```
    while(i <10) //R?
```

```
    { cin >> x;
```

```
        if(x>max)
```

```
            max=x;
```

```
        else
```

```
            if(x<min)
```

```
                min=x;
```

```
            i=i+1;
```

```
    }
```

R=($0 \leq i \leq 10$, letti i valori da cin, min e max sono il minimo ed il massimo degli i valori)

POST ??

correttezza del while:

- 1) condizione iniziale : R vale la prima volta che si arriva a valutare la condizione di permanenza
- 2) invarianza:
 $R \ \&\& \ \text{cond_di_permanenza} \ \langle \text{corpo} \rangle \ R$
- 3) condizione d'uscita: $R \ \&\& \ \neg(\text{cond_di_permanenza}) \Rightarrow \text{POST}$

altra soluzione con inizializzazione:

(ii) $\text{max} = \text{MIN_INT}$ e $\text{min} = \text{MAX_INT}$

```

main() //PRE
{
    int x, min=MAX_INT, max=MIN_INT, i=0;
    while(i <10)
    { cin >> x;
      if(x>max)
        max=x;
      else
        if(x<min)
          min=x;
        i=i+1;
      }

    cout <<"min="<<min<<" max="<<max<<endl;}
POST ??

```

```
int x, min=MAX_INT, max=MIN_INT, i=0;
```

```
while(i <10)
{ cin >> x;
  if(x>max)
    max=x;
  else
    if(x<min)
      min=x;
  i=i+1;}
```

$R=(0 \leq i \leq 10, \text{ letti i valori da cin, min e max sono il minimo ed il massimo degli i valori})$

NON VA

$R'=(0 \leq i \leq 10, \text{ se } i=0 \text{ allora min=MAX_INT e max=MIN_INT, se } i > 0, \text{ allora letti i valori da cin, min e max sono il minimo ed il massimo degli i valori})$????

applichiamo la regola di correttezza del while

1) condizione iniziale : OK

2) invarianza ??? **non va**

CORREZIONE ? ce la suggerisce R' !!

**dobbiamo fare in modo che entrambi min e
max siano a posto.**

```
int x, min=MAX_INT, max=MIN_INT, i=0;
```

```
while(i <10) // R'  
{ cin >> x;  
  if(x>max)  
  {max=x;  
    if (i==0) min=x;  
  }  
  else  
  if(x<min)  
  {min=x;  
    if(i==0) max=x;  
  }  
  i=i+1;}
```

INUTILE ??



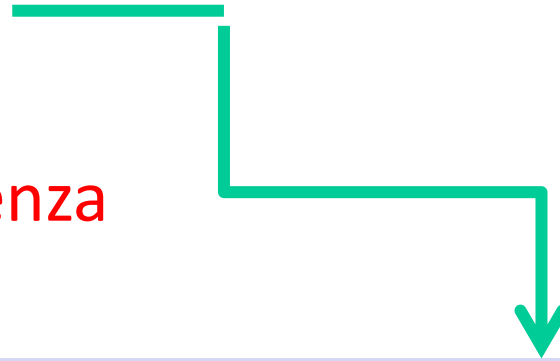
NUOVO Esercizio:

PRE=(cin contiene intero $x > 0$)

POST=(esponente è il min esponente
tale $2^{\text{esponente}} \geq x$)

```
int potenza=1, esponente=0, x;  
cin >> x;
```

```
while (potenza < x )  
    aumenta esponente  
    calcola prossima potenza
```



invariante

$R = (\text{potenza} = 2^{\text{esponente}}, x \geq 2^{\text{esponente}-1})$

POST=(esponente è il min esponente tale
 $2^{\text{esponente}} \geq x$)

```
int x,potenza=1, esponente=0;  
cin >> x;  
while( x > potenza) R = invariante  
{  
    esponente = esponente + 1;  
    potenza=potenza*2;  
}  
cout<< "l'esponente e'=" << esponente<<endl;
```

POST= (esponente è il minimo esponente
tale che $2^{\text{esponente}} \geq x$)

Nuovo esercizio:
si legge sequenza di caratteri numerici
fino a leggere 'a'

esempio: '3' '2' '6' 'a'

vogliamo calcolare l'intero 326

PRE=(cin contiene $c_1 \dots c_k$ 'a', $k \geq 0$, c_j
carattere numerico $j \in [1, k]$)

programma ?

POST=(calcola NUM($c_1 \dots c_k$))

NUM('2' '3' '1')=231 NUM()=0

inizio :

leggi un carattere

se è diverso da 'a', fai i conti e vai all'inizio

altrimenti fine

fai i conti ?

se abbiamo letto

$c_1c_2c_3$ abbiamo $\text{NUM}(c_1c_2c_3)$ e se c_4 ok dobbiamo ricavare $\text{NUM}(c_1c_2c_3c_4)$

come ??

$$\text{NUM}(c_1c_2c_3c_4) = 10 * \text{NUM}(c_1c_2c_3) + \text{NUM}(c_4)$$

```
char q; int num=0, n=1;  
IN>>q;
```

```
R=(num=NUM(c1..c(n-1)), q=cn)
```

```
while(q != 'a')  
{  
    num=num*10+(q-'0');  
    IN >> q;  
    n++;  
}
```

Esercizio: leggere valori interi da IN e
scriverli su OUT

al più 10 valori, ma

fermarsi con due 0 consecutivi **senza
scriverli**

$PRE = (IN = b_1 \dots b_k, k > 9 \text{ o contiene due } 0 \text{ consecutivi})$

programma ?

$POST = (OUT = \text{Prefix}(b_1 \dots b_k))$

$\text{Prefix}(b_1 \dots b_k) = b_1 \dots b_{10}$ se $b_1 \dots b_{10}$ non contiene due 0 consecutivi, altrimenti, se $j < 9$ e $b_1 \dots b_j$ non contiene 00, allora
 $\text{Prefix}(b_1 \dots b_j 00 \dots) = b_1 \dots b_j$

entro nel ciclo se non ho letto due 0 consecutivi e non ho ancora letto 10 valori

-nel ciclo leggo X e **devo distinguere** se X è 0 o no

e devo ricordare se al giro precedente ho letto uno 0 o no

idea: **due booleani per distinguere**

-al giro precedente ho letto uno zero o no

-nei 2 giri precedenti ho letto due zeri o no

invariante R

1. $0 \leq n \leq 10$
2. letti $b1 \dots bn$
3. $(!uno0 \ \&\& \ !due0) \Rightarrow (OUT = b1 \dots bn)$
4. $due0 \Rightarrow (bn-1 = bn = 0 \ \&\& \ OUT = b1 \dots bn-2)$
5. $(uno0 \ \&\& \ !due0) \Rightarrow (bn = 0 \ \&\& \ OUT = b1 \dots bn-1)$


```

int X, n=0;
bool uno0=false, due0=false;
while(! due0 && n < 10)
{ IN >> X; n++;
  if(X==0)
    if(uno0)
      due0=true;
    else
      uno0=true;
  else
    {if(uno0)
      {OUT<< 0 <<' '; uno0=false;}
    OUT << X << ' ';
    }
  if(uno0 && !due0) OUT << 0;
}

```