

CPU e programmazione (Parte 3)

Esecizio “modulo”

Scrivere un programma che, dato un numero reale X , calcola il **valore assoluto** $|X|$ e lo pone nel numero reale AbsX.

Quindi: **se** $X \geq 0$, pone il valore X in AbsX, **altrimenti** lo cambia di segno (ad es. sottraendolo a 0) prima di porlo in AbsX.

Esecizio “modulo-read”

Scrivere un programma che **legge un reale** X dalla tastiera e **stampa** sul video il **valore assoluto** $|X|$. Quindi:

1. legge X dalla tastiera,
2. **se** $X \geq 0$ lo scrive direttamente sul monitor **altrimenti**,
3. lo cambia di segno (ad es. sottraendolo a 0) prima di scriverlo sul monitor.

```
ZeroF: FLOAT 0;  
X: FLOAT -234.43;  
AbsX: FLOAT;  
  
LOAD R0 ZeroF;  
LOAD R1 X;  
FCOMP R1 R0;  
BRGE maggiore;  
FSUB R0 R1;  
STORE R0 AbsX;  
STOP;  
maggiore: STORE R1 AbsX;  
STOP;
```

RAM

ZeroF:
X:
Absx:

CPU

R0:
R1:
RC:

```

ZeroF: FLOAT 0;
X: FLOAT;
AbsX: FLOAT;

READ INP X;
LOAD R0 ZeroF;
LOAD R1 X;
FCOMP R1 R0;
BRGE maggiore;
FSUB R0 R1;
STORE R0 AbsX;
BRANCH stampa;

```

```

maggiore: STORE R1 AbsX;
BRANCH stampa;

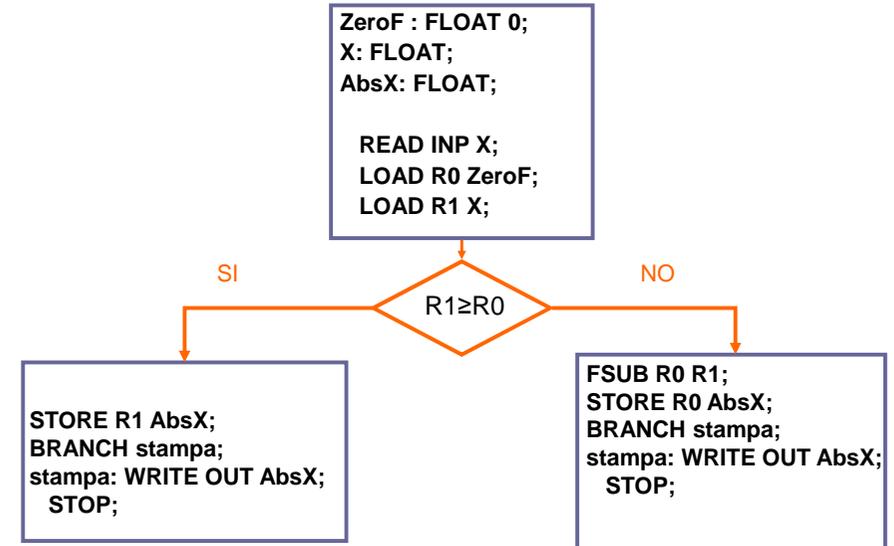
```

```

stampa: WRITE OUT AbsX;
STOP;

```

FLOWCHART



Esercizio “potenza”

Leggere un **reale x** ed un **intero positivo n** e calcolare la **potenza x^n**

Questo esempio utilizza una struttura di controllo di grande importanza: **il ciclo** (o iterazione)

```

X: FLOAT 3;
N: INT 4;
Ris: FLOAT ;
Uno: INT 1;
Unofl: FLOAT 1.0;

```

```

READ INP X;
READ INP N;
LOAD R0 Uno;
SUB R0 R0;
LOAD R1 Uno;
LOAD R2 X;
LOAD R3 N;
LOAD R4 Unofl;

```

R0 = 0 intero
R1 = 1 intero
R2 = X reale
R3 = N intero
R4 = 1 reale

R0 = 0 intero R2 = X reale R4 = 1 reale
 R1 = 1 intero R3 = N intero

```

Ciclo: COMP R3 R0;
          BREQ Esci;
          FMUL R4 R2;
          SUB R3 R1;
          BRANCH Ciclo;
Esci: STORE R4 Ris;
          WRITE OUT Ris;
          STOP;
  
```

$$R4 = X^{N-R3}$$

$$R4 = X^N$$

```

X: FLOAT 3;
N: INT 4;
Ris: FLOAT ;
Uno : INT 1;
Unofl: FLOAT 1.0;
LOAD R0 Uno;
SUB R0 R0;
LOAD R1 Uno;
LOAD R2 X;
LOAD R3 N;
LOAD R4 Unofl;
  
```

X:	3,0
N:	4
Ris:	?
Uno:	1
Unofl:	1,0
R0:	0
R1:	1
R2:	3,0
R3:	4
R4:	1,0
RC:	?

All'inizio del ciclo: $R4 = X^{N-R3}$

```

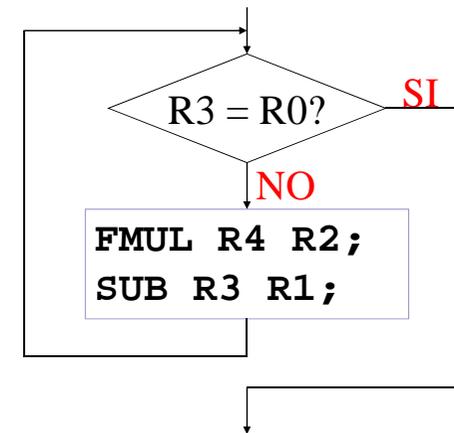
Ciclo: COMP R3 R0;
          BREQ Esci;
          FMUL R4 R2;
          SUB R3 R1;
          BRANCH Ciclo;
Esci: STORE R4 Ris;
          STOP;
  
```

X:	3,0
N:	4
Ris:	81,0
Uno:	1
Unofl:	1,0

R0:	0
R1:	1
R2:	3,0
R3:	0
R4:	81,0

Durante il ciclo: $R4 = X^{N-R3}$

Alla fine del ciclo: $R3=0$ ed $R4 = X^N$ **RC:** 0

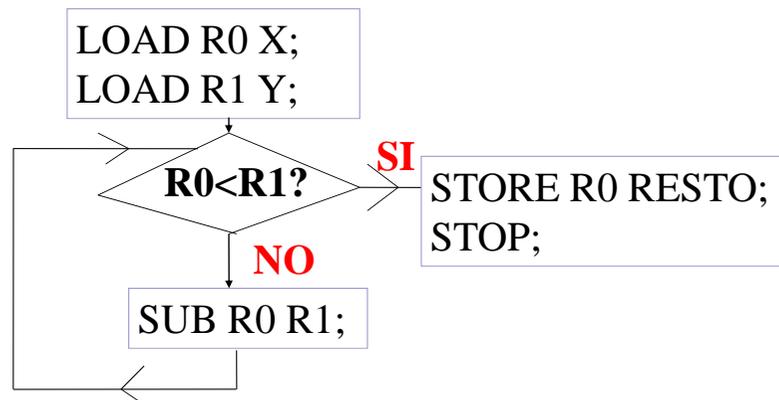


ciclo o iterazione

Esercizio “resto”

Calcolare il resto della divisione intera di due numeri usando solo la sottrazione

```
X: INT 22;  
Y: INT 5;  
RESTO: INT ;  
  
LOAD R0 X;  
LOAD R1 Y;  
ciclo: COMP R0 R1;  
BRLT fine;  
SUB R0 R1;  
BRANCH ciclo;  
fine: STORE R0 RESTO;  
STOP;
```



ciclo o iterazione

Esercizio “max-min”

- Scrivere un programma che legge N reali e ne restituisce il massimo e il minimo. Quindi:
 1. legge N (intero) dalla tastiera,
 2. se $N \leq 0$ termina l'esecuzione del programma, altrimenti
 3. legge gli N numeri e mantiene due registri uno contenente il Max trovato fino a quel punto e uno contenente il Min
 4. ogni nuovo numero letto viene confrontato con il Max e il Min correnti
 5. al termine della lettura degli N numeri vengono scritti in output il Max e il Min

Esercizio "max-min"

- Scrivere un programma che legge **N reali** e ne restituisce **il massimo e il minimo**.

Suggerimento:

Per aggiornare un registro Ri in modo che contenga il dato in Rj:

```
FSUB Ri Ri;  
FADD Ri Rj;
```

```
ZERO: INT 0;  
UNO: INT 1;  
MIN: FLOAT;  
MAX: FLOAT;  
X: FLOAT;  
N: INT;  
  READ INP N;  
  LOAD R0 ZERO;  
  LOAD R2 N;  
  COMP R0 R2;  
  BRLT OK;  
  STOP;  
OK : READ INP X;  
  LOAD R3 X;  
  LOAD R4 X;  
  LOAD R1 UNO;
```

```
CICLO: SUB R2 R1;  
  COMP R2 R0;  
  BRLE ESCI;  
  READ INP X;  
  LOAD R5 X;  
  FCOMP R5 R3;  
  BRGE NONMIN;  
  FSUB R3 R3;  
  FADD R3 R5;  
NONMIN: FCOMP R5 R4;  
  BRLE NONMAX;  
  FSUB R4 R4;  
  FADD R4 R5;  
NONMAX: BRANCH CICLO;  
ESCI: STORE R3 MIN;  
  STORE R4 MAX;  
  WRITE OUT MIN;  
  WRITE OUT MAX;  
  STOP;
```