Corsi di Laurea in Biologia e Biologia Molecolare
Università di Padova, AA 2012/2013

Accumulatore
Vettori

Preparazione per l'esame

www.stoianov.it

ACCUMULATORE

- Se abbiamo istruzioni per sommare (moltiplicare, ecc) due numeri scalari, come possiamo applicare la stessa operazione su più numeri ?
- Soluzione: definire una variabile *accumulatore* **A** ed applicare la stessa operazione tra A e ogni valore in questione.

- **Algoritmo**: dati i ***n*** valori (X*1*, X*2*, … X*n*)
  1) inizializzare l'accumulatore **Y**
  2) applicare l'operazione tra l'accumulatore e ogni valore X*i*, memorizzando il risultato nell'accumulatore.

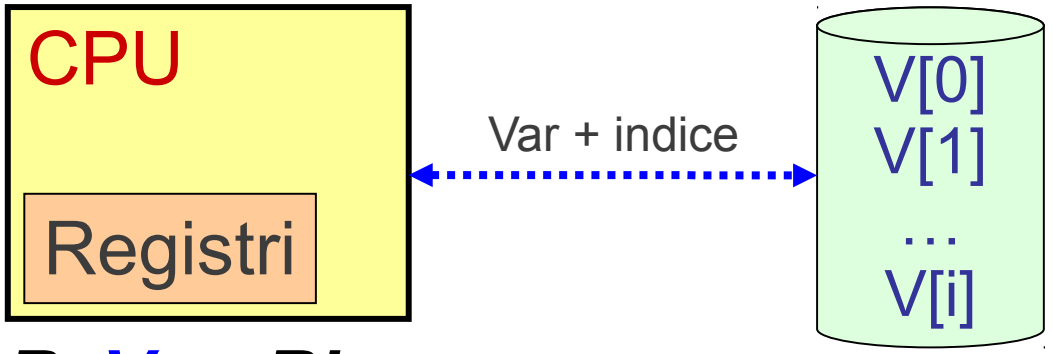
- **Inizializzazione**:
  Metodo A: "azzerare" per accumulare poi tutti gli elementi *Xi*
  - per *sommare / moltiplicare*, inizializzare con 0 / 1
  - per *max* / *min*, inizializzare con un numero piccolo / grande.

  Metodo B: inizializzare l'accumulatore con il primo valore *X1*, per accumulare poi con tutti i valori tranne il primo.

Strutture dati: **Vettori**

- Per poter elaborare più valori, occorre organizzarne in una **struttura dati**. L'efficienza degli algoritmi dipende dalla struttura utilizzata.
- La struttura più semplice è il **vettore,** nel quale ***N*** valori sono memorizzati in ***N*** successive celle *(struttura lineare)*.
- Un vettore **X** (o **X[ ]**) è localizzato nella RAM partendo da un certo indirizzo (del vettore **X**).
- Ogni elemento del vettore **X** viene riferito tramite un **indice *i*** che varia da **0** a ***N***-1 (elemento ***i***: **X[i]**)
- L'indice ***i*** di ogni elemento **X*i*** determina il suo indirizzo relativo rispetto l'indirizzo del vettore ***X***, (***i*** * *byte-per-codificare-un-valore-Xi*)
- L'indirizzo assoluto (nella RAM) di ogni elemento **X*i*** è uguale all' *indirizzo del vettore* **X** più *l'indirizzo relativo di* ***Xi***.

Riferimento a dati vettoriali



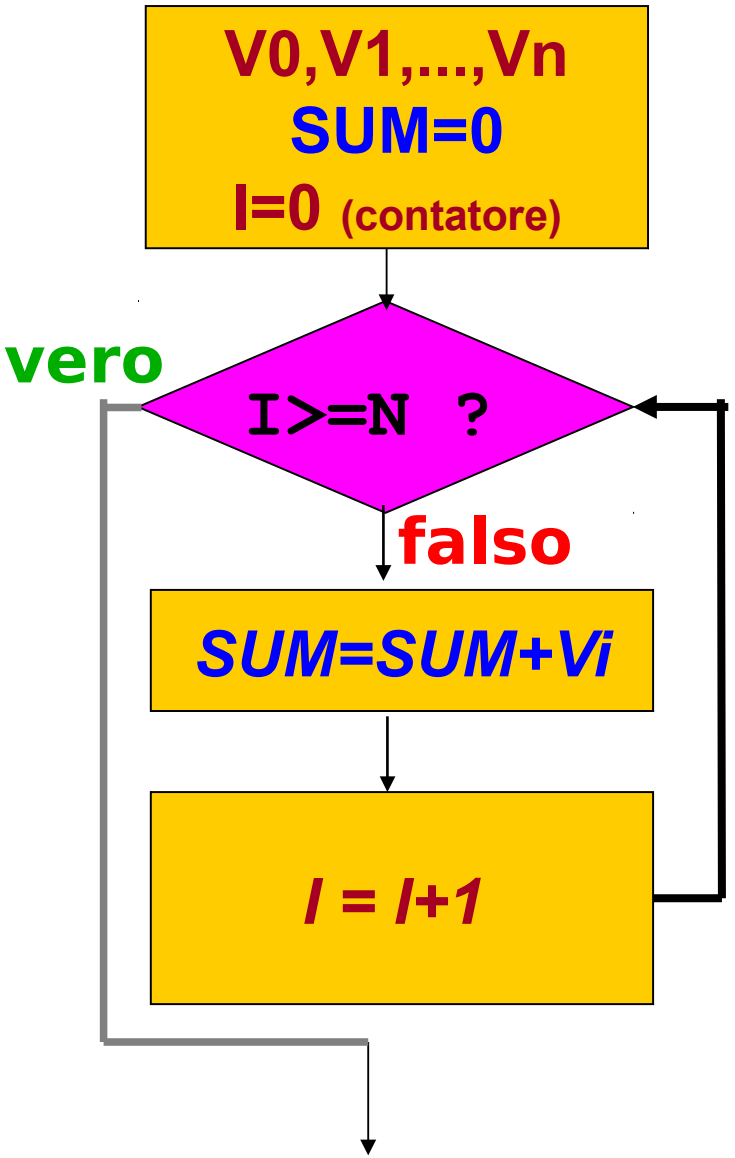
- LOAD ***R***, ***Var***, ***Ri***
  LOAD ***R***, ***Var***, ***i***
  carica nel registro-dati ***R*** il contenuto del elemento **[Ri]/i** della variabile ***Var*** (l'indice = contenuto del registro ***Ri***)

- STORE ***R***, ***Var***, ***Ri***
  STORE ***R***, ***Var***, ***i***
  copia il contenuto del registro dati ***R*** nel elemento **[Ri]/i** della variabile **Var**

In ***microcalc***, ogni variabile potrebbe essere un vettore (massimo 20 elementi):

```
V: INT 10 15 20 25 30;      - definisce i valori iniziali di un vettore V (i=0,1,2,3,4)
LOAD R1 V 4;                - carica in R1 il 5° elemento di V
STORE R1 V 8;               - imposta il valore del 9° elemento di V
```

SOMMA VETTORIALE



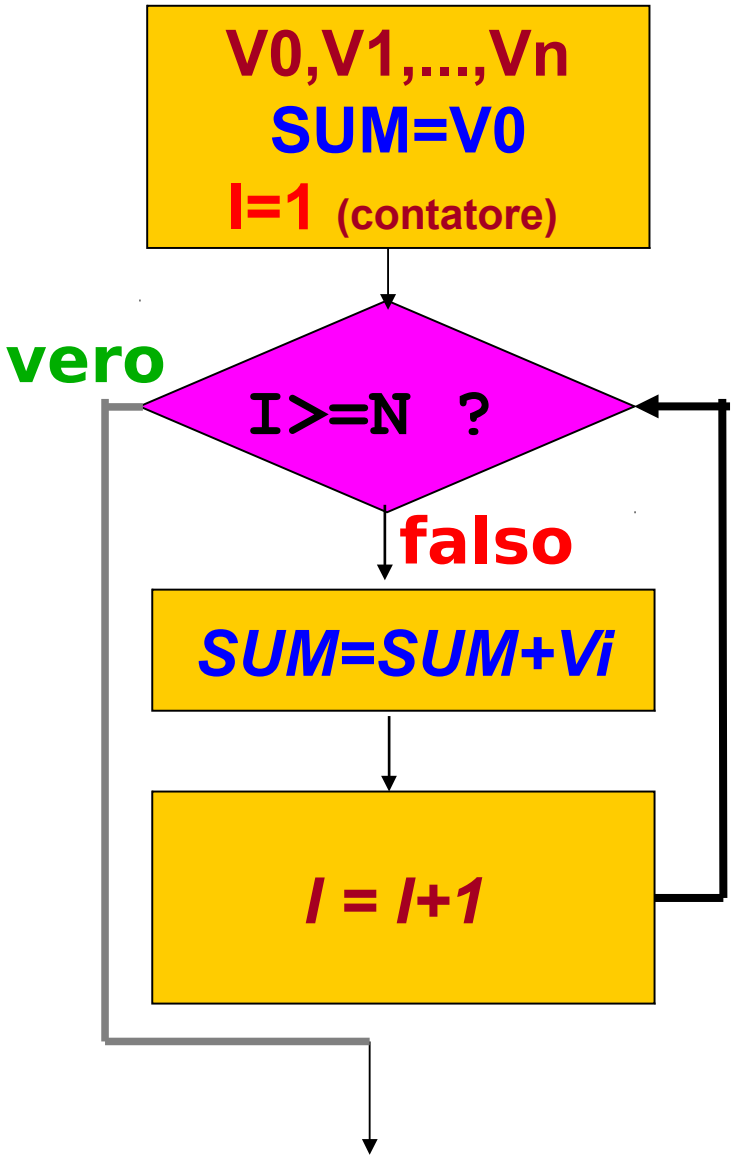

```
V: INT 3 10 2 -1 -4; Vettore V
N: INT 5;    lunghezza di V
S: INT;      Accumulatore - Somma

LOAD R0 0;   Indice i=0
LOAD R1 N;   Numero elementi in V
LOAD R2 0;   Acc=0

Testa: COMP R0 R1; i-N
BRGE Post;   se i>=N, fine ciclo
LOAD R3 V R0; V[i]
ADD R2 R3;   Acc=Acc+V[i]
ADD R0 1;    i=i+1;
BRANCH Testa;

Post: STORE R2 S; S=Acc
STOP;
```

SOMMA VETTORIALE (2)



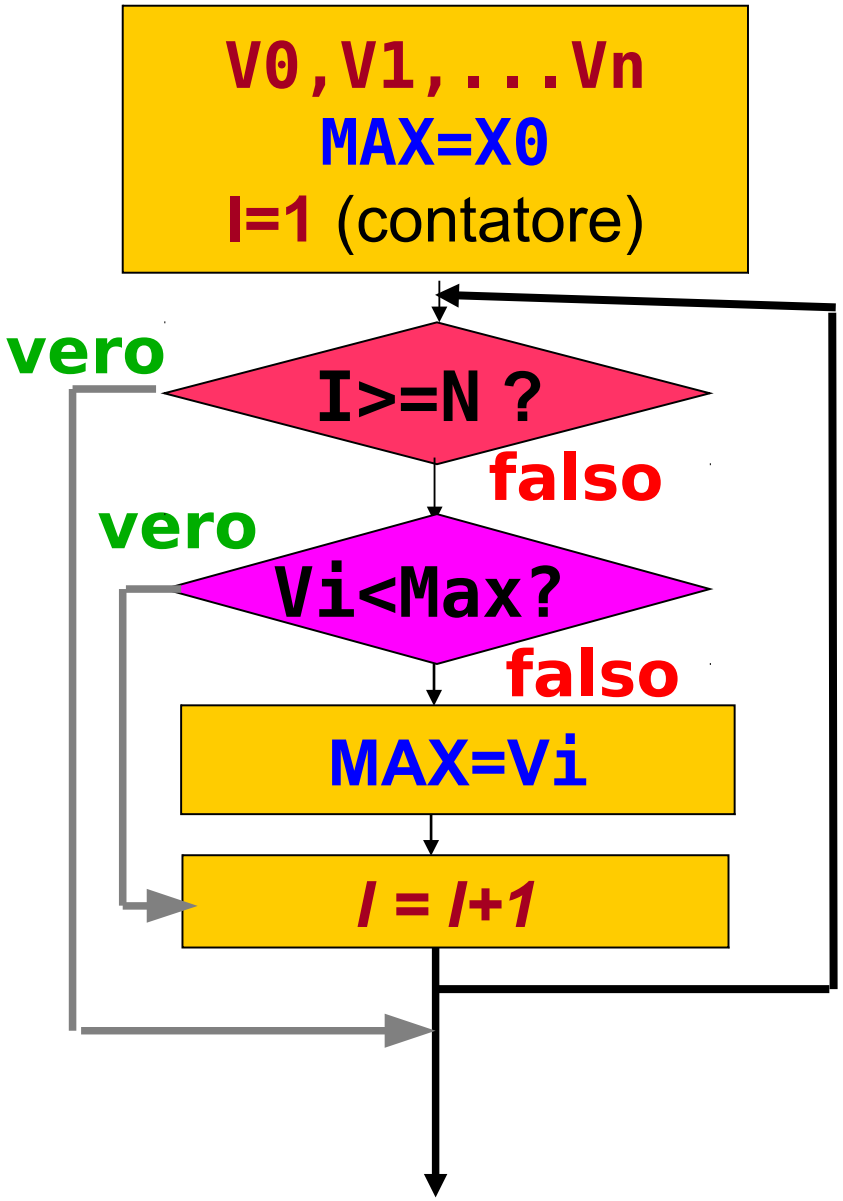

```
V: INT 3 10 2 -1 -4; Vettore V
N: INT 5;  lunghezza di V
S: INT;    Accumulatore - Somma

LOAD R0 1; Indice i=0
LOAD R1 N;
LOAD R2 V 0; Acc=V[0]

Testa: COMP R0 R1; i-N
BRGE Post;  se i>=N, fine ciclo
LOAD R3 V R0; V[i]
ADD R2 R3;  Acc=Acc+V[i]
ADD R0 1;   i=i+1;
BRANCH Testa;

Post: STORE R2 S;
STOP;
```

VALORE MASSIMO




```
V: INT 3 10 2 -1 -4; Vettore V
N: INT 5;  lunghezza di V
M: INT;    Accumulatore - Max

LOAD R0 1; Indice i=0
LOAD R1 N;
LOAD R2 V 0; Max=V[0]

Testa: COMP R0 R1; i-N
BRGE Post;  se i>=N, fine ciclo
LOAD R3 V R0; R2=V[i]
COMP R3 R2; Xi<Max?
BRLT Next;  Yes; next
LOAD R2 R3; No: Max=Xi
Next: ADD R0 1;  i=i+1
BRANCH Testa;

Post: STORE R2 S;
STOP;
```

Preparazione per l'esame

(solo parte programmazione)

info@stoianov.it

Domande con risposte multiple. Indicare con croce solo le risposte corrette. Le risposte sbagliate penalizzano la valutazione della domanda stessa. **(1 punto per domanda)**

1. La memoria RAM:
- contiene solo i dati di un programma
- serve a dare accesso veloce ai dati
- è un circuito costruito da varie porte logiche
- non perde il contenuto se non viene alimentata.

2. **Un algoritmo con cicli**
- ha almeno un salto non-condizionale
- ha almeno un salto condizionale
- ha almeno due salti condizionali e un salto non-condizionale
- utilizza almeno tre variabili

3. **Un programma in codice CPU:**
- codifica un algoritmo in modo comprensibile per il programmatore
- è eseguibile da un qualunque calcolatore digitale
- include almeno un'istruzione di controllo
- non necessariamente ha un corrispondente diagramma di flusso

Domande con risposte aperte **(xx punti per domanda)**

1. A che cosa serve la memoria RAM ?

2. Descrivere i varie tipi di elaborazioni iterativi.

3. Descrivere il ruolo dei diagrammi di flusso nel processo di programmazione.

Programmazione (esempio A) **(5 punti)**

Le variabili ***D1*** e ***D2*** contengono i diametri di due cerchi. Calcolare la superficie cumulativa dei due cerchi, restituendo il risultato in una variabile ***S***.

(I) Fare il diagramma di flusso di un algoritmo che risolve il compito. [2 punti]

(II) Scrivere un programma per il simulatore *micorocalc* che corrisponde a questo algoritmo [2.5 punti].

(III) Verifica la correttezza dell'algoritmo, calcolando a mano (aprossivamente), la somma *S* se *D1*=2, *D2*=6 [0.5 punto]

```
Qui sotto trovi le primi e l'ultima istruzione del programma:
D1: INT 2;  % Il diametro del primo cerchio
D2: INT 6;  % Il diametro del secondo cerchio
S: INT;     % La somma

LOAD R0 D1; % R0 contiene il diametro del primo cerchio
…
STOP;       % Fine del programma. La variabile S contiene la somma
```

Programmazione (esempio B) **(5 punti)**

Le variabili ***M*** e ***N*** sono due numeri interi. Calcolare la somma dei numeri da ***M*** a ***N*** e restituirla in una variabile ***S***.

(I) Fare il diagramma di flusso di un algoritmo che risolve il compito. [2 punti]

(II) Scrivere un programma per il simulatore *micorocalc* che corrisponde a questo algoritmo [2.5 punti].

(III) Verifica la correttezza dell'algoritmo, calcolando a mano la somma *S* se *M*=10 e *N*=13 [0.5 punto]

```
Qui sotto trovi le primi e l'ultima istruzione del programma:
M: INT 10; % Valore iniziale per la sommatoria
N: INT 13; % Valore finale per la sommatoria
S: INT;    % La somma dei numeri da M a N (compresi)

LOAD R0 M; % M = R0
LOAD R1 N; % N = R1
…
STOP;      % Fine del programma. La variabile S contiene la somma
```