

A.A. 2004-2005

Esercitazioni Programmazione 1

Silvia Crafa e Claudia Faggian

Scopo delle esercitazioni

Rispondere a: **FAQ**

- Come scrivere un programma?
 - Da dove comincio?
 - E' possibile *scrivere progr. corretti*?
- Come leggere un programma?
- Come faccio a passare l'esame?

1-3

PRIMA di scrivere una sola linea di codice:

1. analizza il problema (sei sicuro di capire?)
2. risolilo.

Messaggio subliminale:

*E' piu' faticoso correggere un programma
che progettarlo bene*

Come disegnare un programma

**Programmare e' un processo creativo.
Ma...**

Progettare un programma (indicazioni)

In due fasi:

- **Problem Solving**
"che metodo risolve il problema"?
- **Implementazione**

1-6

Problem Solving

- Analizzare e capire:

- Quali sono gli input ?
Chi sono, da dove vengono?
- Qual e' l'output?
Cosa, come lo fornisco

- Sviluppare una soluzione:
il metodo (algoritmo)

Si ma... come?

- Di che variabili ho bisogno?

1-7

Tecniche di sviluppo

- Sviluppo top-down

- ZOOM: descrivi il programma in modo sempre piu' dettagliato

- Dividi in sottoproblemi

- Pseudo-codice

1-8

PSEUDO-CODICE

Se il voto e' maggiore di 18
stampa "Ammesso"
altrimenti stampa "Non sufficiente".

"CODICE":

```
int voto;
```

```
...
```

```
if (voto >= 18)
```

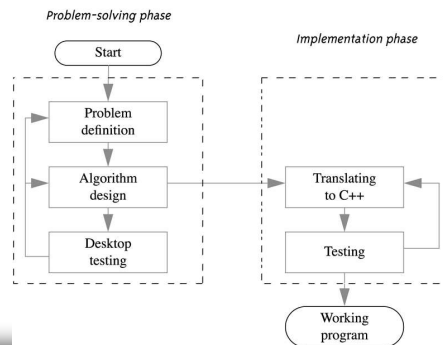
```
    cout << "Ammesso";
```

```
else
```

```
    cout << "Non sufficiente";
```

1-9

Program Design Process



1-10

Evita l'errore del novellino!!

(passare direttamente alla codifica,
pensando di guadagnare tempo)

L'esperienza mostra:
il processo in due fasi produce
un programma che *funziona correttamente*
piu' in fretta

1-11

Come capire un programma

Saper leggere per
saper scrivere

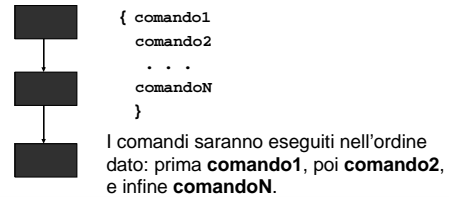
Due domande fondamentali
per capire un programma
(le stesse che si pone il compilatore)

- Qual e' il **prossimo comando** da eseguire?
- Quali sono il **valore** e il **tipo** delle variabili ?

1-13

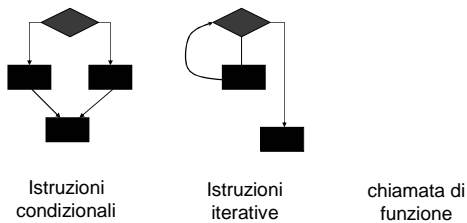
Il flusso di controllo

- Il flusso di controllo è l'ordine con cui i comandi in un programma vengono eseguiti.
- La modalità più semplice è quella **sequenziale**:



1-14

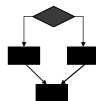
Modificare il flusso di controllo



1-15

If else

Un esempio



```

if (voto >= 18)
    cout << "Ammesso";
else
    cout << "Non sufficiente";

```

1-17

Scrivere un programma che gestisce l'ammissione all'orale.

Per essere ammessi all'orale, bisogna ***passare lo scritto e aver consegnato il progetto.***

```

if (voto >= 18 && consegnato) // condizione
    cout << "Ammesso all'orale " ;
else
    cout << "Non ammesso";

```

1-18

Se ammesso,
stampare anche il voto
(voto scritto + bonus test)

```
if (voto>=18 && consegnato) // condizione
{
    // blocco di istruzioni
    cout << "Ammesso all'orale ";
    voto = voto + bonus;
    cout << "con voto " << voto;
}
else
    cout << "Non ammesso";
```

1-19

Se voto < 18, stampa "non sufficiente"
Altrimenti, guarda se il progetto e' stato consegnato:
Se si, stampa "ammesso all'orale"
Se no, stampa "manca il progetto"

// comandi if_else innestati

```
if (voto < 18)
    cout << "voto non sufficiente"<<endl;
else
{
    if (consegnato)
        cout << "ammesso all'orale\n";
    else
        cout << "manca il progetto\n";
}
```

1-20

Come prendere decisioni

espressioni booleane

Pari o dispari?

Se **y** è **pari** allora **y%2** ha valore 0

Se **y** è **dispari** allora **y%2** ha valore 1

- Scrivi un'espressione booleana che testi se y è pari
- Scrivi un'espressione booleana che testi se y è dispari e strettamente positivo
- Scrivi un'espressione booleana che testi se y è multiplo di 5 o di 7

1-22

Cicli: WHILE

Es. uso di ciclo:
algoritmo per fare la spesa

- Prendo un carrello ed entro al supermercato
- Fino a che ci sono prodotti sulla lista
 - Acquisto un prodotto della lista
 - Lo barro
- Passo alla cassa

1-24

Le regole d'oro del ciclo sano



1. Come **inizia** il ciclo?
2. Cosa **fa avanzare** il ciclo?
3. Cosa **ferma** il ciclo?

Ricordati **sempre** di
chiudere il ciclo!

1-25

Calcolare la media dei voti sul mio libretto

Ho dato 3 esami:

• Inizializzo le variabili → voto
somma =

Leggo 3 voti e li sommo → Ripeto 3 volte:
leggo un voto
somma = somma + voto

• Calcolo la media → Divido la somma per 3

1-26

Calcolare la media dei voti sul mio libretto

voto
somma = 0

voto
somma = 0
contatore =

Ripeto 3 volte:
leggo voto
somma = somma + voto

Fisso un contatore.
while (contatore < 3)
leggo voto
somma = somma + voto
aggiorno contatore

1-27

Codificare e' facile!

```
int cont=0;           //contatore
int voto;
int somma=0;

// ad ogni ripetizione,
// a cosa corrispondono somma e cont?
while (cont < 3) {
    cout << "voto?\n";
    cin >> voto;
    somma = somma + voto;    // somma dei voti letti
    cont= cont+1;           // ho letto cont voti
};

cout << "la media e' "
    << (somma /3 ) << endl;

media_cont.c
```

1-28

Riassumendo

WHILE:
ripeti fintanto che la condizione resta VERA

è VITALE assicurarsi che
ad un certo punto la condizione diventa FALSA



usare i /* commenti */
per indicare come si esce dal ciclo

1-29

Come disegnare un ciclo

while (condizione) CORPO

Devi disegnare 3 cose:

1. Il **corpo** del ciclo : le istruzioni da ripetere (ricorda di far avanzare il ciclo!)
2. **Inizializzazione** delle variabili
3. Le condizioni per **terminare**

1-31

Modi per terminare

condizione controllata da un **contatore**:

- esegue un numero predeterminato di volte

condizione controllata da un **evento**:

- esegue fino ad un evento (**sentinella**), che termina il ciclo.

*Importante sapere
cosa si sta usando:
anche mix dei due!!*

1-32

Calcola la media dei voti
per un libretto qualunque

uso una sentinella

- Inizializza variabili

Leggi voti fino a
sentinella

→

```
while (voto!=sentinella)
    leggi voto
    somma = somma+voto
```

- Calcola media

1-33

Calcola la media dei voti
per un libretto qualunque

uso una sentinella

- Inizializza variabili

Leggi voti fino a
sentinella

→

```
leggi voto
while (voto!=sentinella)
    somma = somma+voto
    leggi voto
```

- Calcola media

→ Ci serve un contatore dei
voti letti

1-34

```
int cont, voto, somma;
cont=0;
somma =0;

cout << "voto?\n";
cin>>voto;

// uscita: somma contiene la somma dei voti letti
while (voto!=0) {
    cont = cont+1;           //quanti voti ( ) letti
    somma = somma + voto;   //somma voti letti
    cout << "voto?\n";
    cin >> voto;
}

//mi metto al sicuro da possibile divisione per 0
if (cont !=0)
    cout << "la media e' " << (somma /cont) << endl;
```

1-35

Ed ora un po' di pepe...

Indovina quanti fagioli
ci sono nel vaso

```
rand();
```

Es:

```
int numero_fagioli = rand();
```

```
//qual e' ora il valore della variabile numero_fagioli ?
```

1-37

Testa o croce?

*rand() produce
un numero a caso tra 0 e (almeno) 32767*

Come possiamo simulare il lancio di una moneta?

0/1 per testa/croce

int moneta=rand()%2

1-38

Compiti per casa

La morra cinese

Scrivere un programma che simuli il gioco della morra cinese

REGOLE DEL GIOCO

☞ ogni giocatore sceglie un simbolo tra "forbici", "carta" e "sasso".

☞ I due simboli scelti dai giocatori vengono quindi confrontati:

forbici battono carta:  vince 

carta batte sasso:  vince 

sasso batte forbici:  vince 

1-40

Compiti per casa

- Dove:

www.math.unipd.it/~claudia/Prog1

- CHE EDITOR ? Nel CD cygwin trovate Xemacs, emacs.
Qlq. editor con highlights va benissimo. Per una scelta:
www.thefreecountry.com/programming/editors.shtml

- Ogni file inizia con nome dell'autore/degli autori

1-41