

INFORMATICA

CORSO DI INFORMATICA DI BASE

ANNO ACCADEMICO 2015/2016

DOCENTE: SARRANTONIO ARTURO

PROGRAMMA

- **Aspetti fondamentali della programmazione con riferimento ad un linguaggio di programmazione: astrazione sui dati e uso di tipi (in particolare: i costruttori di tipo array, struct e puntatori),**
- **strutture di controllo dell'esecuzione (condizionali, di selezione, iterative).**
- **Sottoprogrammi: funzioni e procedure.**
- **Progetto top-down dei programmi (anche con l'impiego di sottoprogrammi). Passaggio dei parametri nei sottoprogrammi, variabili locali e globali, ricorsione.**
- **Gestione dei file.**

PROGRAMMAZIONE

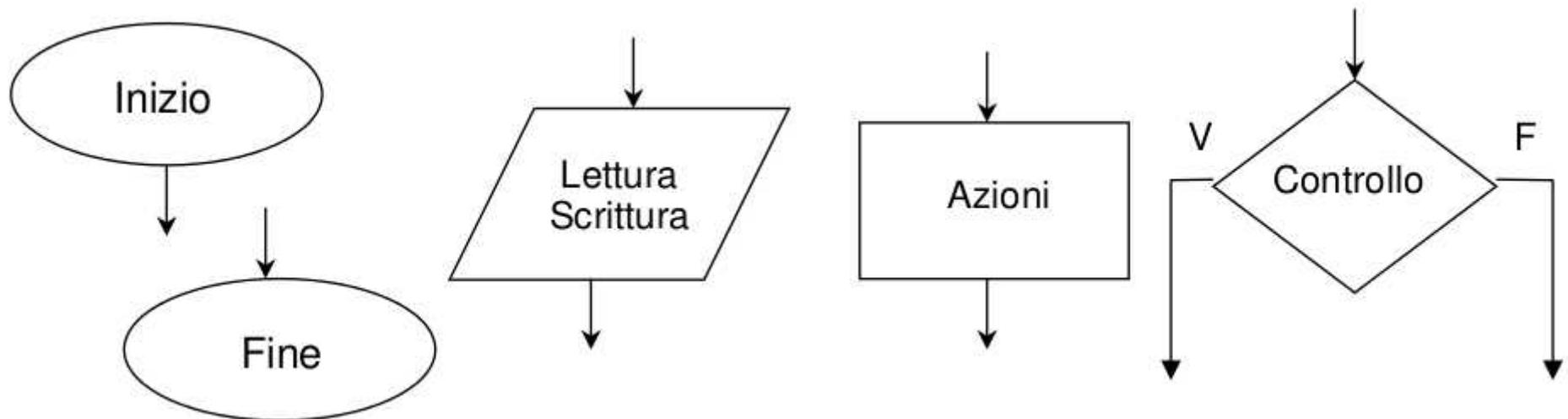
- Programmazione: È l'attività con cui si predispone l'elaboratore ad eseguire un particolare insieme di azioni su particolari dati, allo scopo di risolvere un certo problema.
- Affinché un problema sia perfettamente risolubile è necessario che la sua definizione sia esatta e completa. Ci sono alcuni problemi che non sono risolvibili attraverso l'uso del calcolatore.

PROGRAMMAZIONE

- I problemi che possono essere risolti con un calcolatore possono essere di varia natura:
- Sommare due numeri interi;
- Ordinamento crescente di numeri
- Dati a e b , risolvere l'equazione $ax + b = 0$
- Dati due numeri trovare il maggiore;
- Gestire consumi energetici
- Trovare le radici di una funzione $f(x)$;
- Approssimare e interpolare funzioni.

LINGUAGGI DI PROGRAMMAZIONE

- Prima di procedere alla scrittura di un programma partendo da un algoritmo si prepara lo schema a blocchi.
- I possibili tipi di blocco sono i seguenti



PROGRAMMAZIONE

- Un programma traduce un diagramma a blocchi (che identifica un algoritmo)
- Quando si programma occorre seguire determinate strutture (si parla pertanto di **PROGRAMMAZIONE STRUTTURATA**)

PROGRAMMAZIONE

- Linguaggio di riferimento: C
- Esempi di programma in C

PROGRAMMAZIONE

- Un programma C deve contenere, nell'ordine:
- una parte contenente le direttive per il compilatore;
- l'identificativo main
- due parti racchiuse da `{ }`
 - una parte dichiarativa;
 - una parte esecutiva.
 - La parte dichiarativa contiene le seguenti sezioni:
 - □ dichiarazione delle costanti;
 - □ dichiarazione delle variabili .

PROGRAMMAZIONE

- Linguaggio di riferimento: C
- Esempi di programma in C

*/*programma che letti due numeri a terminale ne stampa la somma*/*

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
int X,Y;
```

```
scanf("%d%d",&X,&Y);
```

```
printf("%d",X+Y);
```

```
}
```

PROGRAMMAZIONE

- Struttura di un programma C:
- Titolo
- `#include <stdio.h>` (libreria da includere)
- `#include` (libreria da includere)
- `Main()` (programma principale)
- Routine

PROGRAMMAZIONE

- Le librerie sono insiemi di comandi che possono/non possono essere inclusi nel programma
- Hanno comandi specifici relativi a determinati argomenti
- esempi

PROGRAMMAZIONE

Esempi di routine in C

```
if ((Eta >=6) && (Eta <=14))
printf("%s","scolare");
else {
printf("%s","Non scolare");
printf("%d",Eta);
};
```

PROGRAMMAZIONE

- Per l'esecuzione di un programma all'interno del `main()` o delle routine ci deve essere una parte di dichiarazione di variabili
- Le variabili sono di diversa tipologia

PROGRAMMAZIONE

- I tipi di dati che è possibile utilizzare in un programma C appartengono a due categorie:
 - tipi di dati semplici
 - tipi di dati strutturati.
- Il linguaggio C prevede quattro tipi di dati semplici:
 - char (carattere); 1 byte
 - int (intero) 2 byte
 - float (reale) 4 byte
 - double (reale in doppia precisione) 8 byte

PROGRAMMAZIONE

- Esempio: Leggere e stampare una stringa

```
main(){  
char s[20];  
printf("Inserisci una stringa: ");  
scanf("%s", s);  
printf("Hai inserito la stringa: %s", s);  
}
```

- La libreria per la gestione delle stringhe è

```
#include <string.h>
```

PROGRAMMAZIONE

- TIPO STRUCT

- Il costruttore di tipi struct permette di definire dei tipi strutturati aggregando informazioni (ovvero variabili) eterogenee tra loro correlate.

- •Una struttura viene definita mediante il costruttore struct nel seguente modo:

```
struct nome_struttura
{
tipo_1
nome_campo_1;
tipo_2
nome_campo_2;
.....
tipo_N
nome_campo_N;
};
```

- dove tipo_i e nome_campo_i indicano, rispettivamente, il tipo e il nome dell'i-esimo campo della struttura.

PROGRAMMAZIONE

- TIPO STRUCT

- Per individuare i campi occorre individuare variabile e campo

paz.nome

paz.cognome

etc...

PROGRAMMAZIONE

- TIPO STRUCT Esempio programma di acquisizione di una struttura fatta di 10 studenti

```
#include <stdio.h>
#define N 20
struct studente {
char nome[20];
char cognome[20];
long matricola;
int
nmaterie;
float media;
};
main()
{int presenti,ind;
struct studente S[N];
printf("Indicare il numero di studenti \n");
scanf("%d",&presenti);
for(ind=0;ind<presenti;ind++)
{printf("Cognome: ");
scanf("%s",S[ind].cognome);
printf("Nome: ");
scanf("%s",S[ind].nome);
printf("Matricola: ");
scanf("%ld",&S[ind].matricola);
printf("Numero di materie: ");
scanf("%d",&S[ind].nmaterie);
printf("Media: ");
scanf("%f",&S[ind].media);
}
for(ind=0;ind<presenti;ind++)
{printf("Cognome: %s\n",S[ind].cognome);
printf("Nome: %s\n",S[ind].nome);
printf("Matricola: %s\n",S[ind].matricola);
printf("Numero di materie: %d\n",S[ind].nmaterie);
printf("Media: %f\n",S[ind].media);
}
}
```

PROGRAMMAZIONE

gestione dei files

- Le periferiche del computer archiviano i files con una struttura ad albero “rovesciato”
- La radice è la directory principale, le altre directory sono i rami e le foglie sono i files.