



Università degli Studi di Cagliari
Corso di Laurea in Ingegneria Biomedica

ELEMENTI DI INFORMATICA

https://www.unica.it/unica/page/it/gianluca_marcialis

A.A. 2021/2022

Docente: **Gian Luca Marcialis**

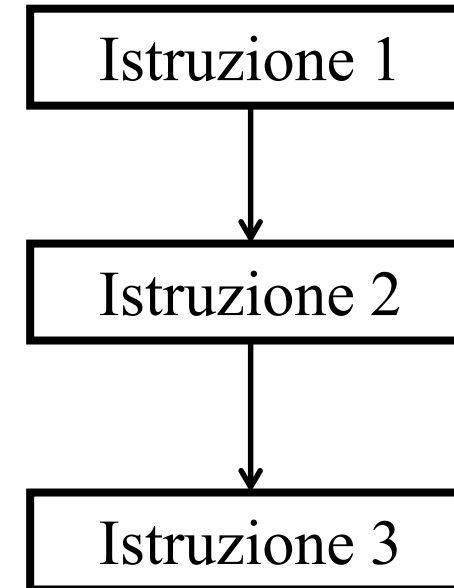
LINGUAGGIO C
Strutture di controllo

Sommario

- Introduzione
- Il costrutto `if...else`
- Il costrutto `while`
- Altri costrutti
 - Il costrutto `switch`
 - Il costrutto `for`
 - Il costrutto `do-while`
- Altre istruzioni di controllo del flusso

Introduzione

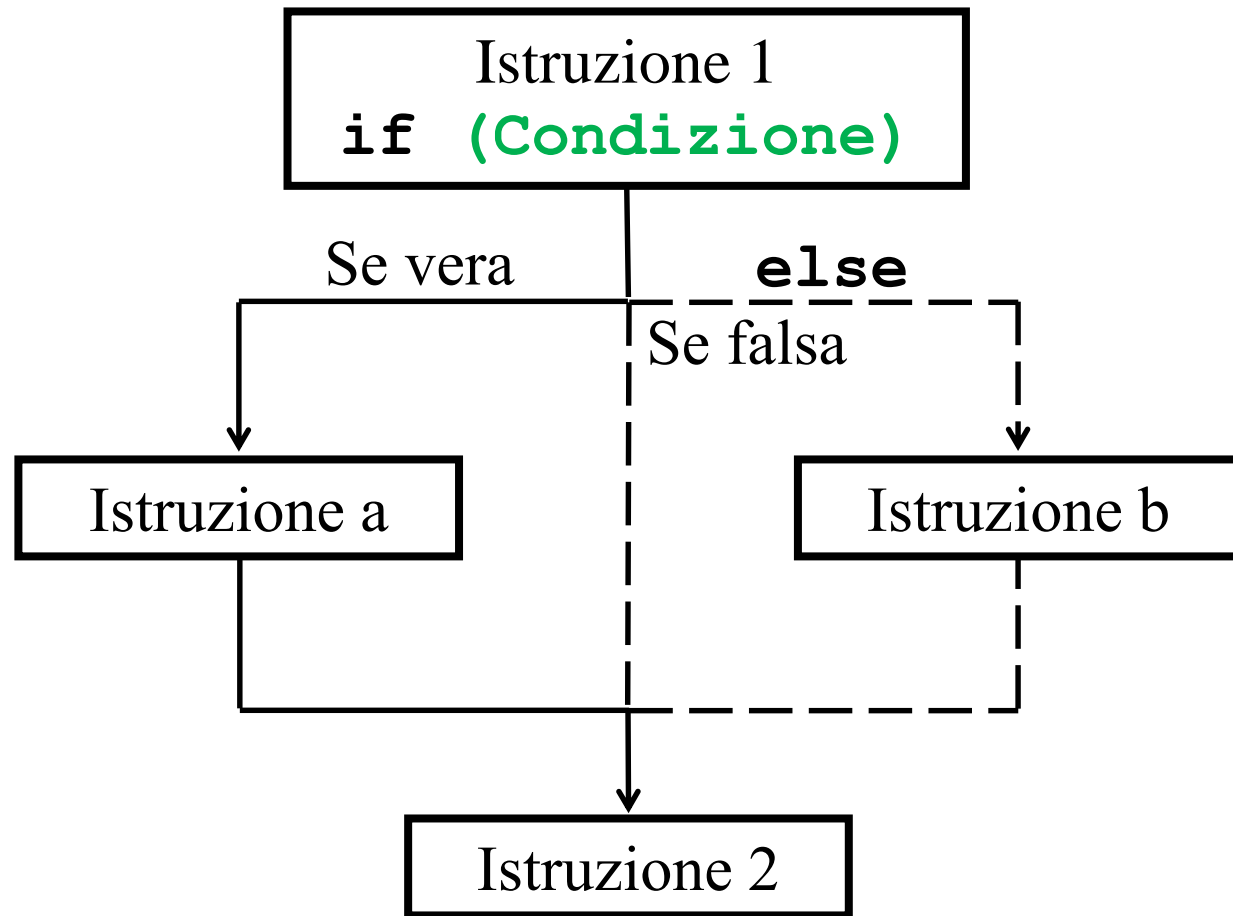
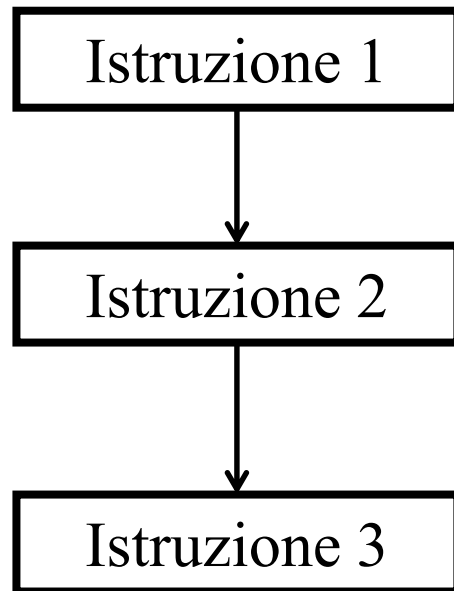
- L'esecuzione di un programma è, in linea di principio, puramente **sequenziale**
- Nella scrittura di algoritmi c'è però la necessità, spesso, di alterare il normale flusso di esecuzione delle istruzioni, ricorrendo ad opportuni costrutti detti **strutture di controllo**
- Le strutture di controllo **alterano** il flusso di esecuzione sequenziale
- Le strutture di controllo sono di due tipi:
 - Condizionali (la maggior parte)
 - Incondizionali



Esigenze fondamentali

- Ramificazione o «branching»
 - Separare blocchi di istruzioni mutuamente esclusivi in base al verificarsi di una condizione
 - Costrutti: `if-else`, `switch-case`
- Cicli, iterazioni o «loop»
 - Ripetizione di un blocco di istruzioni in base al verificarsi di una condizione
 - Costrutti: `while`, `for`, `do-while`
- In entrambi i casi la condizione è rappresentata da una variabile booleana o da un'espressione booleana

Ramificazione: `if-else`



Sintassi e semantica di if-else

- Modello in pseudo-linguaggio: Se-altrimenti

```
if (Condizione) /*Se Condizione è vera*/  
{  
    /*esegui le istruzioni nelle graffe*/  
}  
[else /*altrimenti, se Condizione è falsa*/  
{  
    /*esegui le istruzioni in queste graffe*/  
}] ➔ ramo else non obbligatorio
```

Il costrutto `if...else`

- Nel costrutto `if...else` il termine indicato come `Condizione` è un'espressione booleana che è interpretata come **vera** se il suo valore è diverso da 0, altrimenti è **falsa**
- `Condizione` può essere quindi espressa come
 - combinazione degli operatori booleani noti (AND, OR, NOT) di variabili booleane (ovvero intere ma con significato booleano) oppure
 - espressioni di confronto determinate, ad esempio, dal valore corrente assunto da variabili numeriche o caratteri, oppure
 - altre forme più complesse (che vedremo più avanti)

Il primo esempio

- Scrivere un programma C che, ricevendo da tastiera due valori interi, segnali qual è il maggiore fra essi.

```
/*Programma che stampa il maggiore tra due valori interi*/
#include <stdio.h>
int main()
{
    int x, y, differenza, maggiore;

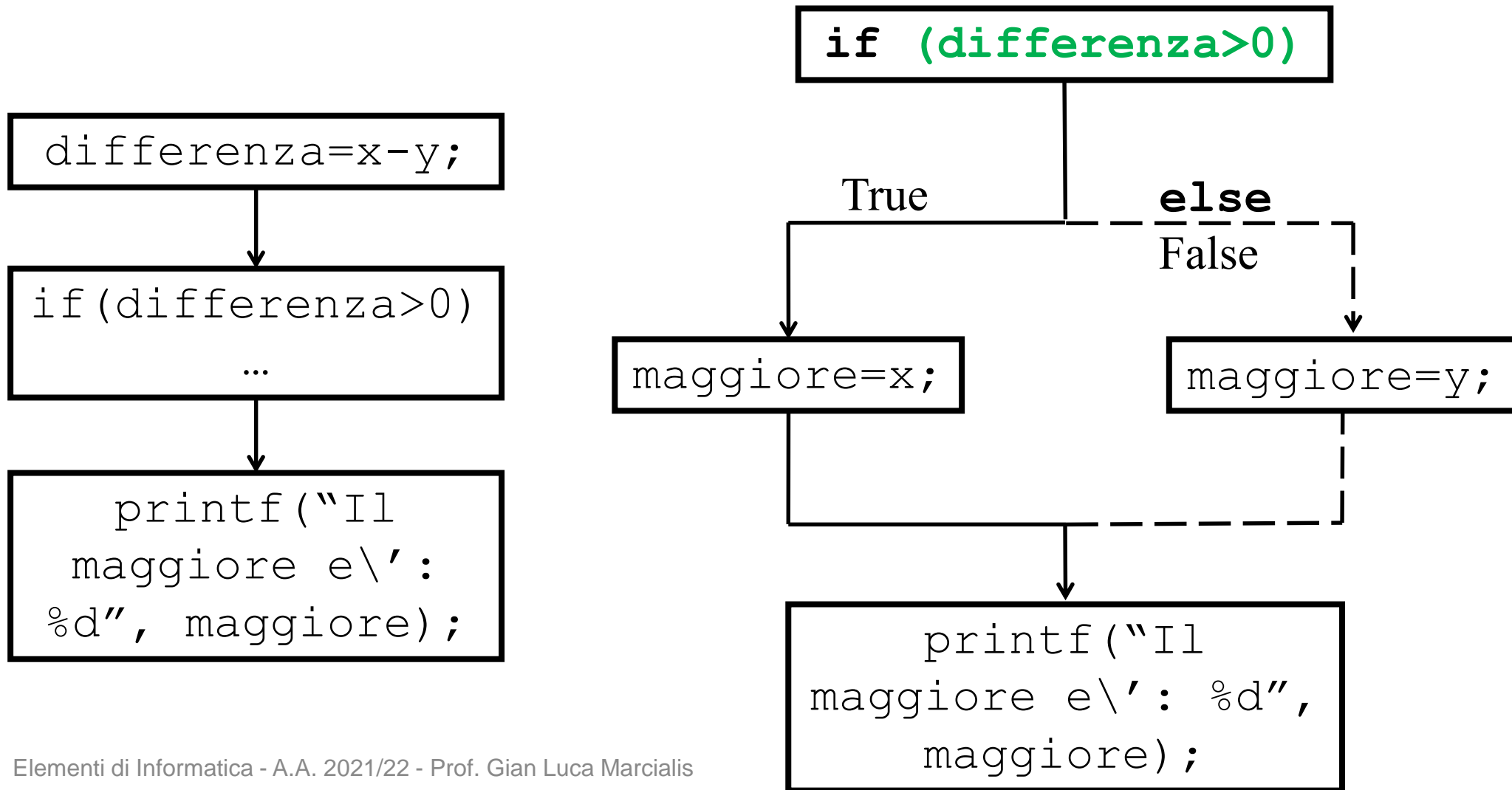
    scanf("%d %d",&x, &y); /*Lettura dei dati*/

    /* Elaborazione dei dati */
    differenza = x - y;
    if (differenza > 0)      /*se il valore di differenza è maggiore di zero...*/
        maggiore=x;         /*assegna a maggiore il valore di x*/
    else                    /*altrimenti*/
        maggiore=y;         /*assegna a maggiore il valore di y*/

    printf("Il maggiore e\' : %d", maggiore); /*Stampa del risultato*/

    return 0;
}
```

Modello grafico nell'uso di `if`



Esercizio

- Scrivere un programma C che, ricevendo in ingresso un valore intero non negativo, compreso tra 0 e 33, stampi a video:
 - “Valore non corretto” se il valore inserito è al di là del range consentito;
 - “Hai passato l’esame e puoi registrarlo!” se il valore inserito è compreso fra 21 e 33;
 - “Dovrai sostenere l’orale se vuoi passare”, se il valore è compreso fra 16 e 20;
 - “Mi dispiace.. non potrai nemmeno sostenere il prossimo scritto...”, se il valore è minore od uguale a 15.

Soluzione

```
/*Programma per conoscere l'esito dell'esame di Fondamenti di Informatica*/
#include <stdio.h>
int main()
{
    int voto;

    printf("Inserire il voto.\n");
    scanf("%d",&voto);

    if((voto<0)|| (voto>33))
        printf("Valore non corretto.\n");
    else
        if((voto>20)&&(voto<=33))
            printf("Hai passato l'esame e puoi registrarlo!\n");
        else
            if((voto>=16)&&(voto<21))
                printf("Dovrai sostenere l'orale se vuoi passare.\n");
            else
                printf("Mi dispiace... non potrai nemmeno sostenere il prossimo scritto...\n");

    return 0;
}
```

Donne ed informatica

Giannotti Fosca Home » People

Fosca Giannotti is a director of research of computer science at the Information Science and Technology Institute "A. Faedo" of the National Research Council, Pisa, Italy. Fosca Giannotti is a pioneering scientist in mobility data mining, social network analysis and privacy-preserving data mining. Fosca leads the Pisa KDD Lab - Knowledge Discovery and Data Mining Laboratory, a joint research initiative of the University of Pisa and ISTI-CNR, founded in 1994 as one of the earliest research lab on data mining. Fosca's research focus is on social mining from big data: smart cities, human dynamics, social and economic networks, ethics and trust, diffusion of innovations. She is author of more than 300 papers. She has coordinated tens of European projects and industrial collaborations. Fosca is currently the coordinator of [SoBigData](#), the European research infrastructure on Big Data Analytics and Social Mining, an ecosystem of ten cutting edge European research centres providing an open platform for interdisciplinary data science and data-driven innovation. Recently she became the recipient of a prestigious ERC Advanced Grant entitled XAI - Science and technology for the explanation of AI decision making.

Topics:

- Constraint-Based Frequent Pattern Mining
- Mobility Data Mining
- Privacy-Preserving Data Mining
- Complex Network Analysis and Mining
- Data Mining

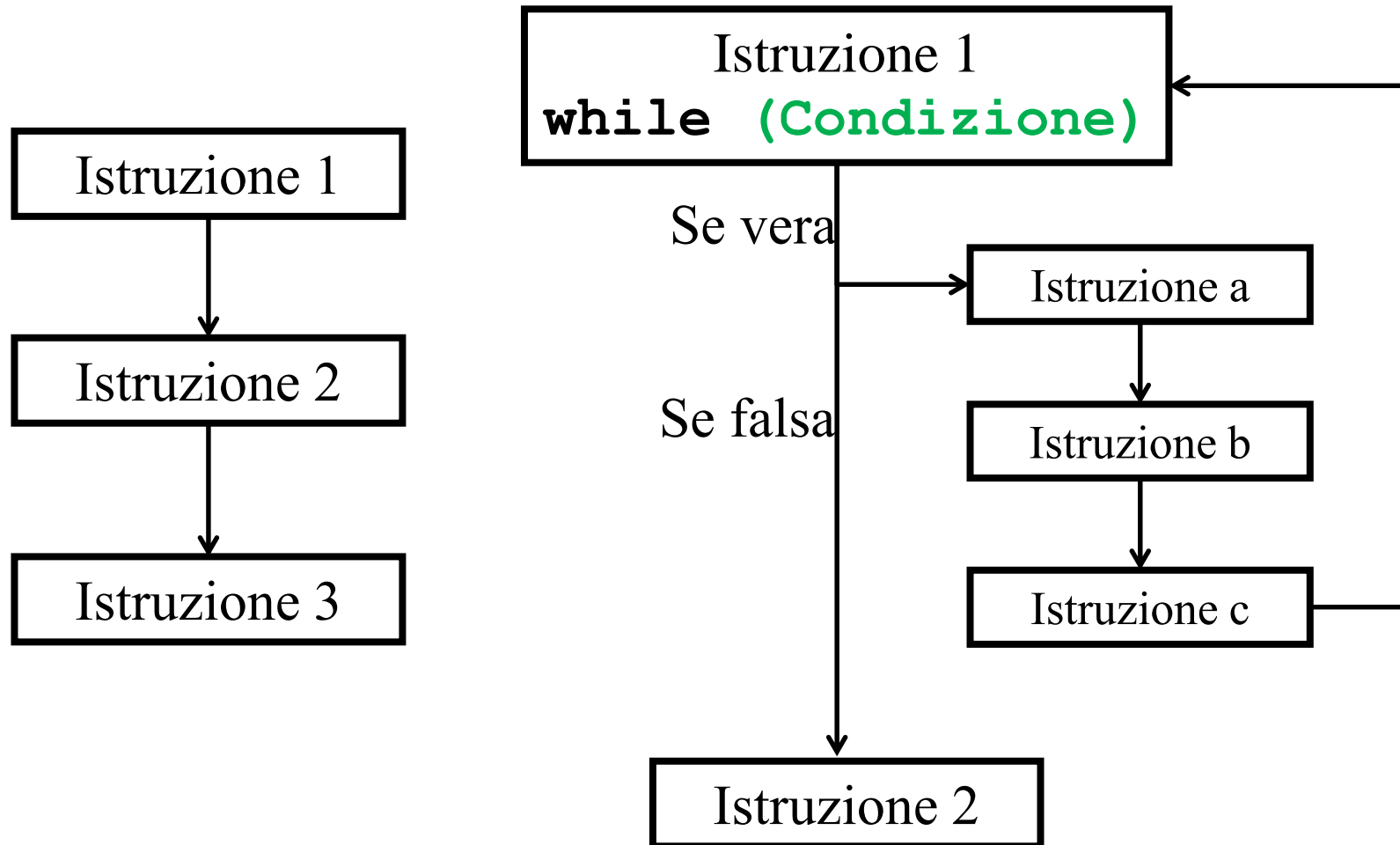
Experience **Activities** **Education**

1982 Master Degree in Computer Science at University of Pisa with 110/100 cum laude

<https://www.youtube.com/watch?v=XxKziMYctPw>

«You got **a choice**. You got **talent**. You got **a chance**. And I don't wanna hear about your environment, what your mama didn't give you, and the **males** won't give you no shot. You got a chance. Use it while you're still alive.» (?)

Cicli o «loop»: while



Sintassi e semantica di `while`

- Modello in pseudo-linguaggio: Ripeti-Finché

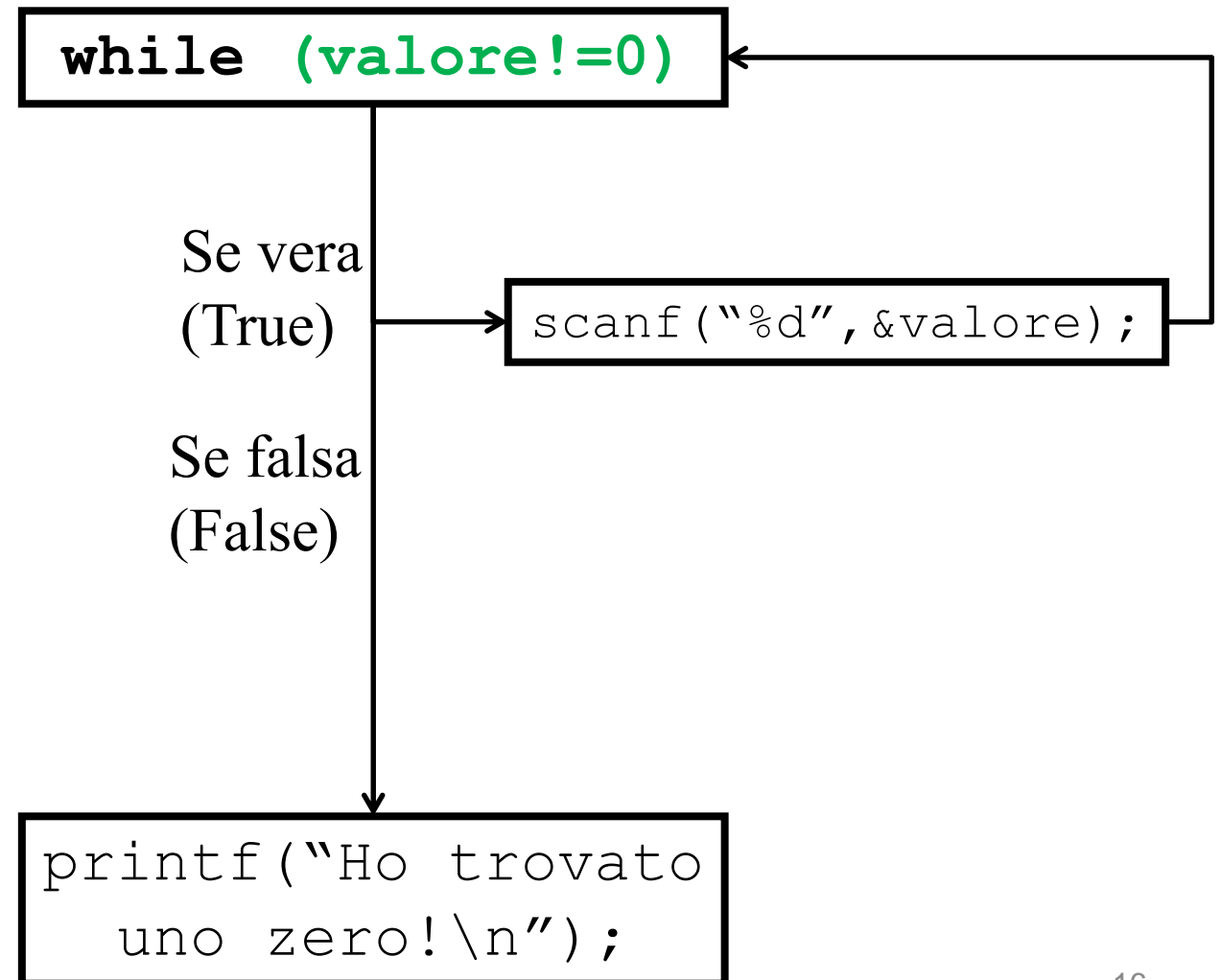
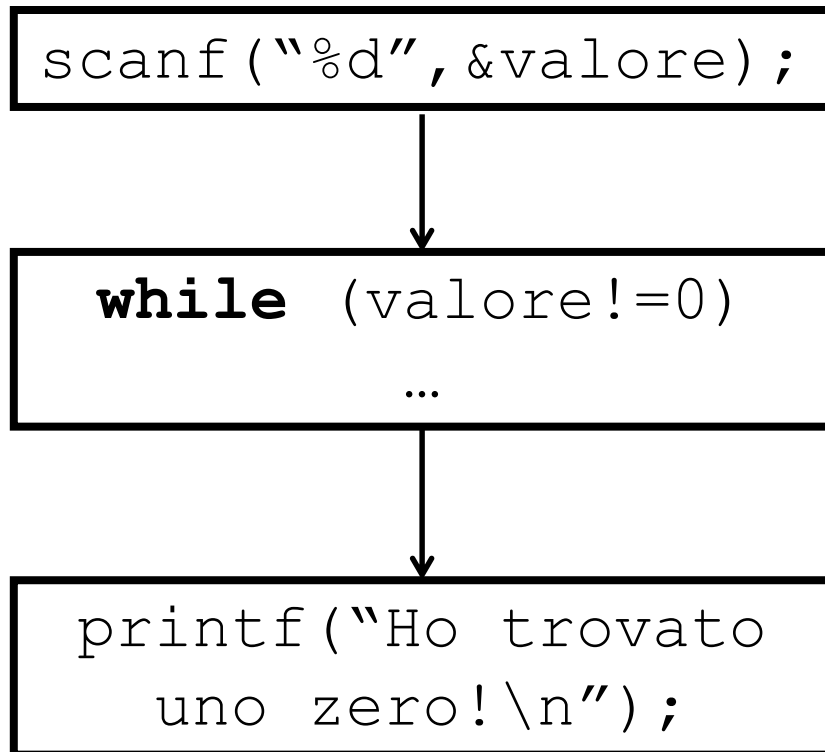
```
while (Condizione) /*Finché Condizione è vera*/  
{  
    /*Esegui le istruzioni tra le graffe*/  
}
```

Esempio

- Scrivere un programma C che legga una sequenza di numeri interi da tastiera e fermi la lettura al primo zero

```
/*Programma che cerca il primo zero*/
#include <stdio.h>
int main()
{
    int valore;
    scanf("%d",&valore);
    while (valore!=0)
        scanf("%d",&valore);
    printf("Ho trovato uno zero!\n");
    return 0;
}
```

Modello grafico nell'uso di `while`



Un esempio più interessante

- Scrivere un programma C che calcoli la media aritmetica di una sequenza di valori reali progressivamente letti da tastiera, finché non viene inserito uno zero

```
/*Programma per la media aritmetica*/
```

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

```
int main()
```

```
{
```

```
    float valore, media;
```

```
    int n;
```

```
    printf("Inserire una sequenza di valori, terminare con 0.\n");
```

```
    scanf("%f",&valore);
```

```
    /*la lettura di un float richiede %f in luogo di %d*/
```

```
    n=0; media=0.0;
```

```
    while (valore!=0.0)
```

```
    {
```

```
        n=n+1;
```

```
        media=media + valore;
```

```
        scanf("%f",&valore);
```

```
    }
```

```
    media = media / ((float)n); /*forza n ad essere rappresentato come float*/
```

```
    printf("La media dei valori inseriti è pari a %f\n",media);
```

```
    return 0;
```

```
}
```

Esercizi su `if` e `while`

- Scrivere un programma C che:
 - legga una sequenza di valori in virgola mobile da tastiera e li memorizzi in un vettore v ;
 - stampi a video tutti i valori di v che risultano maggiori della media aritmetica;
 - memorizzi in un vettore di interi w la parte intera di tutti gli elementi che presentano parte frazionaria strettamente maggiore di 0.5
 - letto un valore intero da tastiera, stampi se il valore è uguale ad uno degli elementi di w .

Altri esercizi

- Scrivere un programma C che legga un numero indefinito di valori interi e si fermi quando sono stati inseriti **più di tre zeri**.
 - Es. una possibile sequenza è 0 1 2 -1 -5 10 0 3 -1 1 0 -1 10 12 0.
- Scrivere un programma C che legga un numero indefinito di bit **in forma di intero** da tastiera e li memorizzi **in una stringa**. Si fermi quando viene inserito il valore -1. Il massimo numero di bit da leggere è 50. Stampi la stringa di bit inserita.
 - Attenzione alla corrispondenza 0/1 nella tabella ASCII!
- Scrivere un programma C che legga 10 valori interi non negativi da tastiera e stampi a video il **massimo** valore letto.

Algoritmi in C: un esempio completo

- Scrivere un programma C che, ricevendo da tastiera un numero intero decimale, lo converta in un valore binario utilizzando **l'algoritmo delle divisioni successive** e lo stampi a video

L'algoritmo delle divisioni successive in “pseudo-codice”

- Ingresso (Input): una sequenza di cifre decimali X (il numero da convertire)
- Uscita (Output): una sequenza di n cifre binarie (il numero convertito) $\{b_{n-1} \dots b_0\}$
- $i = 0$; (operatore “=”: assegnazione)
- **Ripeti**
 - $Q = \text{quoziente di } X/2$;
 - $R = \text{resto di } X/2$;
 - $b_i = R$;
 - $X = Q$;
 - $i = i + 1$;
- **Finché** $Q \neq 0$
 - Nota “ \neq ” significa “diverso da”

Analisi delle “strutture dati” utili

- Ci serviamo di una variabile (intera non negativa) che conservi il valore decimale, chiamiamola X
- Un'insieme di variabili intere che contenga i bit del numero convertito
 - Per questo scopo ci serviamo di un vettore di interi, b , di dimensione massima N predefinita
 - N viene definito attraverso la direttiva `#define`
- Per il calcolo del resto della divisione x/y , che verrà memorizzato in un'altra variabile R utilizziamo l'operatore `%` tra interi
- Terremo anche conto della lunghezza effettiva della stringa di bit memorizzandola nella variabile M ($\leq N$)

Soluzione: parte dichiarativa del codice e lettura di X da tastiera

```
/*Programma per la conversione da decimale a binario*/  
#include <stdio.h>  
#define N 100 /*il numero max di bit è 100*/  
  
int main()  
{  
    unsigned int X;  
    int b[N], R, Q;  
    int M, i;  
  
    printf("Inserire il valore da convertire.\n");  
    scanf("%d",&X);
```

Inizializzazione della variabile b

```
/*Inizializzazione a zero della stringa di bit*/  
/*E' necessaria perché all'atto della dichiarazione di b, i  
valori assegnati a ciascuna componente sono casuali*/
```

```
i=0;  
while (i<N)  
{  
    b[i]=0;  
    i=i+1;  
}
```

Corpo dell'implementazione

```
i=0;
while ( (i<N) && (X!=0) )
{
    Q=X/2;      /*quoziente della divisione*/
    R=X % 2;    /*resto della divisione*/
    b[i]=R;     /*il resto è assegnato a b[i]*/
    X=Q;        /*riaggiorno X*/
    i=i+1;      /*incremento i*/
}
M=i;
```

Controllo della dimensione e stampa del risultato

```
/*Bisogna verificare che N sia stato sufficiente a convertire il valore inserito*/
if(X!=0)
    printf("Il valore inserito non può essere rappresentato con %d bit\n",N);
else /*altrimenti si può stampare il valore convertito*/
{
    printf("Il valore ottenuto richiede %d bit ed è: ",M);

    i=M-1; /*la stampa deve avvenire in ordine inverso*/
    while(i>=0)
    {
        printf("%d",b[i]);
        i=i-1;
    }
}

return 0;
}
```

Esercizio

- Scrivere un programma C che, letta una stringa di caratteri da tastiera di lunghezza massima N, calcoli la lunghezza effettiva della stringa.
- Nell'implementazione si tenga presente che:
 - L'ultimo carattere di una stringa è `'\0'`
 - Una stringa si legge da tastiera con `scanf` usando `%s`

Soluzione

```
/*Programma che calcola la lunghezza di una stringa*/
#include <stdio.h>
#define N 50

int main()
{
    char stringa[N];
    int lunghezza;
    printf("Inserire una stringa di max %d caratteri:\n",N);
    scanf("%s",stringa);
    /*stringa indica la locazione di memoria del primo carattere*/

    lunghezza=0;
    while(stringa[lunghezza]!='\0') lunghezza++;
    /*in C lunghezza++ è lo stesso che scrivere lunghezza = lunghezza +1*/

    printf("Hai inserito la stringa: %s\nLa sua lunghezza è pari a %d\n",stringa, lunghezza);
    return 0;
}
```

Esercizi con i floating point

- Leggere iterativamente una tripla di valori floating point rappresentanti due cateti ed ipotenusa di un triangolo rettangolo. Il programma termina quando alla prima tripla che non rispetta il teorema di Pitagora, dopo aver stampato il numero di triple lette esclusa l'ultima.
- Scrivere un programma che, dopo aver letto da tastiera dieci valori floating point ed averli memorizzati in un vettore, ne estragga e stampi il maggiore.
- Scrivere un programma che risolva l'equazione $ax^2 + bx + c = 0$, letti da tastiera i parametri a, b, c .
 - Libreria `math.h` per il calcolo della radice quadrata: funzione `y=sqrt(x)`.
- Scrivere un programma che, letti 10 valori reali da tastiera, memorizzi in un vettore `v` solo quelli la cui parte frazionaria è strettamente maggiore di 0.5.
- Scrivere un programma che, letto un valore reale da tastiera con solo parte frazionaria (ovvero a parte intera nulla), lo converta in una stringa binaria secondo l'algoritmo delle moltiplicazioni successive.
 - Es. se viene inserito 0.25, il sistema stampa la stringa "0.01"
- Scrivere un programma che, letta una stringa rappresentante la parte frazionaria di un numero binario, lo converta nella corrispondente parte frazionaria reale.
 - Es. se viene inserita la stringa "01" il sistema stampa 0.25.

Il costrutto switch-case

- Alternativa alla `if...else` ed utile per rendere più leggibile un programma in presenza di molteplici “else” in funzione di una certa espressione

```
switch (espressione)
{
    case CASO1: { /*esegui queste istruzioni/*
                break; }
    case CASO2: { /*esegui queste istruzioni/*
                break; }
    ...
    default: { /*esegui queste istruzioni/*
               break; }
}
```

Esempio

- Scrivere un programma C che converta una nota musicale scritta in notazione tedesca (con le lettere dell'alfabeto C, D, E, F, G, A, B, H)

```
/*Programma per la conversione delle note*/
#include <stdio.h>
int main()
{
    char nota;
    printf("Inserire una nota:");
    scanf("%c",&nota);
    switch (nota)
    {
        case 'C': printf("Hai inserito un do\n"); break;
        case 'D': printf("Hai inserito un re\n"); break;
        case 'E': printf("Hai inserito un mi\n"); break;
        case 'F': printf("Hai inserito un fa\n"); break;
        case 'G': printf("Hai inserito un sol\n"); break;
        case 'A': printf("Hai inserito un la\n"); break;
        case 'B': printf("Hai inserito un si bemolle\n"); break;
        case 'H': printf("Hai inserito un si\n"); break;
        default: printf("Non hai inserito alcuna nota\n");
    }
    return 0;
}
```

Esercizio

- Scrivere un programma in C che, letti due numeri float da tastiera x e y, seguiti da un simbolo '+', '-', '*', '/', '^', stampi a video il risultato dell'operazione associata a ciascuno di questi simboli.
- Usare il costrutto **switch-case**.

Il costrutto **for**

- Equivalente al costrutto **while** ma più espressivo nella sintassi
- Molto usato quando si conosce “a priori” la durata di un loop

```
for(stato_iniziale; condizione_di_terminazione;  
    passo_di_iterazione)  
{  
    /*esegui queste istruzioni finché condizione_di_terminazione  
    è vera prima di avere eseguito il passo_di_iterazione*/  
}
```

Esercizio

- Scrivere un programma C che, leggendo da tastiera una stringa di caratteri rappresentanti ciascuno una cifra binaria, lo converta nell'equivalente valore decimale e lo stampi a video
- Note:
 - Il carattere meno significativo è in posizione 0 nella stringa
 - Per l'elevamento a potenza x^y ci serviamo della funzione di libreria `pow(x, y)` contenuta nella libreria `<math.h>`

Ricordiamo l'algoritmo

- $i=0$;
- $X = 0$;
- Ripeti
 - $X = X + b_i * 2^i$
 - $i = i + 1$
- Finché $i < N$

Soluzione

```
/*Programma per la conversione da binario a decimale*/
#include <stdio.h>
#include <math.h>
#define N 100;
int main()
{
    char b[N];
    unsigned int valore, i, cifra;
    valore=0;
    for(i=0; b[i]!='\0'; i++)
    {
        switch (b[i])
        {
            case '0': cifra=0; break;
            default: cifra=1;
        }
        valore=valore+cifra*pow(2,i);
    }
    printf("\nIl valore binario %s convertito risulta essere %d\n",numero,valore);
    return 0;
}
```

Elementi di Informatica - A.A. 2021/22 - Prof. Gian Luca Marcialis

Esercizi con `for`

- Scrivere un programma C che implementi l'algoritmo delle sottrazioni successive per ottenere quoziente e resto della divisione di un numero intero non negativo X con un altro intero non negativo Y e li stampi a video
- Scrivere un programma C che, dati due valori interi non negativi X e Y , calcoli il valore X^Y e lo stampi a video
 - **Senza** usare `pow(X, Y)`
- Riscrivere tutti gli esercizi precedenti usando **`for`** invece di **`while`**

Il costrutto do-while

- Alternativo a `while` (e `for`) con la differenza che la condizione di fine ciclo viene analizzata dopo avere eseguito le istruzioni da iterare

do

```
{  
    /*esegui queste istruzioni*/  
}
```

while (Condizione)

L'istruzione `break`

> Serve per interrompere un ciclo (`while`, `for`, `do-while`) o l'esecuzione delle istruzioni entro il costrutto `switch`

> Esempio:

```
while (1)
{
    scanf ("%d", &x) ;
    if (x==0) break;
}
printf("\nHo trovato uno zero!");
```

Scrivere un programma C che...

- Data una stringa fatta solo di caratteri minuscoli, la trasformi in una stringa di caratteri maiuscoli.
- Data una stringa che esprima un numero da 0 a 5 (esempio "uno", "due"...), stampi a video il valore intero corrispondente.
- Legga un insieme di stringhe da tastiera rappresentanti valori binari come "zero" e "uno", scriva in un vettore di interi la corrispondente sequenza. La lettura si interrompe all'inserimento della parola "fine".
 - Es. inserita la sequenza "zero", "uno", "uno", "zero", "fine", nel vettore di interi ci saranno i valori 0, 1, 1, 0.
 - Si utilizzi la funzione `strcmp(a,b)` che vale 0 se le stringhe `a` e `b` sono uguali. `strcmp` è presente nella libreria `string.h`
 - Esempio: `strcmp(a, "fine")` vale 0 se `a` contiene la stringa "fine".
- Data una stringa, conti il numero di vocali e lo stampi a video per ciascuna vocale.
- Date due stringhe `a` e `b`, verifichi se `a` è uguale a `b`.
 - **Senza usare `strcmp`.**
- Data una stringa che rappresenti una parola, verifichi se la parola data è palindroma.
 - Es. la stringa "osso" è palindroma perché si può leggere da sinistra a destra e da destra a sinistra.
- Data una stringa `s` ed una stringa `ss`, scriva a video se `ss` è una sottostringa di `s`.
 - Es. la stringa "uon" è una sottostringa di "buonanotte".

Esercizio

Scrivere un programma C che legga da tastiera le coordinate x-y di un insieme di punti sul piano cartesiano, e li inserisca in due vettori di variabili dichiarate appositamente da voi. Per inserire un punto è richiesto l'inserimento di una stringa identificativa il nome del punto, ed i valori sul piano ad esso associati. Per esempio:

```
Inserisci dei punti e premi invio ("P0" per fine inserimento):
```

```
P1: 30.1 29.3
```

```
P2: -5.0 6.5
```

```
P3: 10.6 9.2
```

```
P0
```

L'inserimento della stringa "P0" determina la cessazione della lettura dei punti da tastiera.

Il programma calcola il baricentro dei punti inseriti e lo stampa a video. Il massimo numero di punti inseribili da tastiera è 50.

Per confrontare due stringhe si usi la funzione `strcmp(s1, s2)` della libreria `string.h` che restituisce 0 se la stringa `s1` è uguale alla stringa `s2`.

Da vedere dopo «Calcolatori Elettronici»

Rimando all'architettura del calcolatore: azione delle strutture di controllo

- Modulo CPU: alterando il registro che contiene l'indirizzo della istruzione successiva da eseguire (registro PC)
- Ciò determina il fatto che la successiva istruzione non è più quella contigua ma quella indicata dalla struttura

Istruzione corrente
<code>scanf("%d %d",&x, &y);</code>

Registro PC
2

Memoria principale	
Indirizzo	Parola (contenuto)
0	<code>int x, y, z;</code>
1	<code>scanf("%d %d",&x, &y);</code>
2	<code>z=x+y;</code>
...	
1024	<code>return 0;</code>
...	

Per saperne di più

- Ceri, Mandriola, Sbattella, *Informatica – arte e mestiere*, Capp. 3, 6, McGraw-Hill
- Kernighan, Ritchie, *Il linguaggio C*, Cap. 3, Pearson-Prentice Hall