





Laboratorio di Algoritmi e Strutture Dati

Prof. Aniello Murano

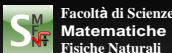
Introduzione al Corso - Il Linguaggio C (I parte)

Corso di Laurea Codice insegnamento Email docente Anno accademico	Informatica 13917 murano@na.infn.it 2007/2008	Lezione numero: 1 Parole chiave: Introduzione, Linguaggio C
--	--	---

[next](#)

27/09/2007



Introduzione al Corso

Informazioni Generali sul Corso:

- Esame: Laboratorio di Algoritmi e strutture dati (6 CFU)
- Libri di testo:
 - [Brian W. Kernighan](#) e [Dennis M. Ritchie](#), "Linguaggio C ", Jackson libri
- Approfondimenti:
 - Al Kelley e Ira Pohl, "C Didattica e Programmazione", Addison Wesley Italia.
 - Dantona e Damiani "Il linguaggio C++ ", Addison Wesley Italia.
- Modalità d'esame: Una prova di laboratorio a gruppi (di tre persone) e una prova scritta.

back
✖
next



Slide 3: Informazioni sul Docente

27/09/2007

Federica

3

Facoltà di Scienze
Matematiche
Fisiche Naturali

Introduzione al Corso

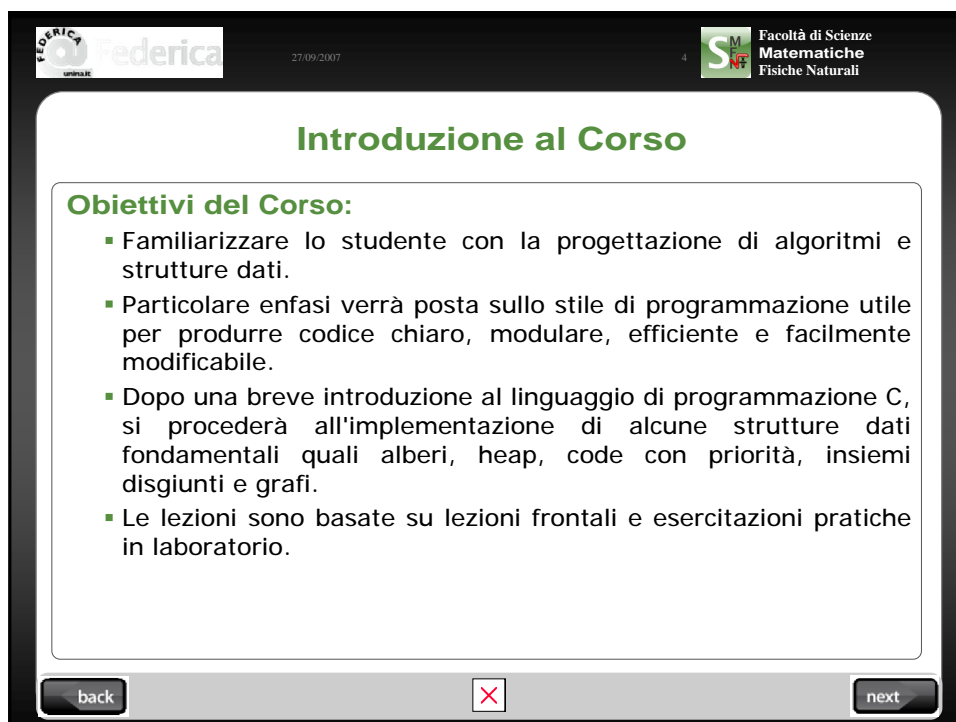
Informazioni sul Docente:

- Prof. Dr. Aniello Murano, ricercatore universitario presso la Sezione di Informatica del Dipartimento di Fisica – Università degli Studi di Napoli “Federico II”
- Sito web: <http://people.na.infn.it/~murano/>
- Ricevimento: Studio OF29b – Edificio del Dip. di Biologia, nei giorni e orari indicati sul sito web del docente
- E-mail: murano@na.infn.it

back

✕

next



Slide 4: Obiettivi del Corso

27/09/2007

Federica

4

Facoltà di Scienze
Matematiche
Fisiche Naturali

Introduzione al Corso

Obiettivi del Corso:

- Familiarizzare lo studente con la progettazione di algoritmi e strutture dati.
- Particolare enfasi verrà posta sullo stile di programmazione utile per produrre codice chiaro, modulare, efficiente e facilmente modificabile.
- Dopo una breve introduzione al linguaggio di programmazione C, si procederà all'implementazione di alcune strutture dati fondamentali quali alberi, heap, code con priorità, insiemi disgiunti e grafi.
- Le lezioni sono basate su lezioni frontali e esercitazioni pratiche in laboratorio.

back

✕

next

Federica 27/09/2007 5 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al Corso

Finalità del Corso:

- Al termine del corso gli studenti dovranno essere in grado di realizzare un progetto completo, comprensivo dei seguenti passi:
 - Analisi del problema
 - Individuazione di una soluzione efficiente
 - Stesura del codice
 - documentazione delle scelte effettuate e del codice prodotto.

back ✖ next

Federica 27/09/2007 6 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al Corso

Programma del Corso:

- Breve Introduzione al C:
 - Origini del C e sue relazioni con altri linguaggi di programmazione, Librerie, Tipi di dati, Espressioni ed istruzioni, Operazioni di input/output, Procedure e funzioni.
- Tecniche di progetto (divide-et-impera, ordinamento, ricerca, selezione) e analisi asintotica
- Implementazione delle seguenti strutture dati:
 - Stack e Code, Liste semplici, doppiamente puntate e circolari, Heap binari e code di priorità, Alberi binari di ricerca.
- Definizione della struttura dati grafo e sua rappresentazione in memoria
- Algoritmi su grafi
 - Creazione, interrogazione e modifica di grafi rappresentati con matrici e liste di adiacenza Visita di grafi (BFS e DFS).

back ✖ next

Federica 27/09/2007 7 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al Linguaggio C (Prima parte)

Cenni storici

Caratteristiche generali

- Struttura di un programma C
- Variabili e costanti

Istruzioni elementari

- Espressioni ed operatori
- Assegnazioni
- Input/output

Istruzioni di controllo del flusso

- Istruzioni condizionali
- Iterazioni

back X next

Federica 27/09/2007 8 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Cenni Storici

- Il linguaggio C è stato sviluppato intorno al 1972 nei Bell Laboratories AT&T americani, da Dennis Ritchie.
- E' nato come linguaggio di sviluppo del Sistema Operativo UNIX.
- Gli antenati del C possono essere riuniti in linea genealogica:
 - **Algol 60** 1960 (Comitato Int.)
 - regolarità della sintassi, struttura in moduli, particolarmente complesso
 - **CPL** (Combined Programming Lan.) 1963 e **BCPL** 1967 (Cambridge)
 - migliorarono le caratteristiche dell'Algol ma non la complessità.
 - **B** 1970 (Thompson)
 - molto legato alla struttura dell'hardware.
 - **C** 1972 (Ritchie)
 - riassume le migliori caratteristiche dei precedenti
- 1986: Objective C (Cox), C++ (Stroustrup)
- 1999: Ultima release dello standard ANSI

back X next

Federica 27/09/2007 9 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Perché programmare in C

- Portabilità del Codice e del Compilatore
- Codice generato molto efficiente
- Facilità di accesso al livello "macchina"
- Interfacciamento completo al S.O. UNIX
- Varietà di operatori di linguaggio
- Strutture dati potenti
- Non complesso (poche keywords)
- Modularità e Riutilizzo

back X next

Federica 27/09/2007 10 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Passi fondamentali:

- Scrivere un codice sorgente in linguaggio C con un **editor**
- Produzione di codice eseguibile utilizzando un **compilatore**
- Dev C++ è un compilatore Free (under the GNU General Public License)

Risorse:
[DevC++](#)
[Html.it](#)

back X next

Federica 27/09/2007 11 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Struttura del Programma Sorgente C

- Un **programma sorgente C** è formato da uno o più blocchi chiamati **funzioni**
- La definizione di una funzione rappresenta la specificazione delle operazioni che dovranno essere svolte all'atto della chiamata.
- La definizione è costituita da due parti:
 - **intestazione dichiarativa**
 - **Corpo**
- L'**intestazione** definisce le regole di interfaccia.
- Il **corpo** specifica le operazioni da eseguire ed è formata da un insieme di istruzioni delimitato utilizzando le parentesi graffe **{ corpo }**

back X next

Federica 27/09/2007 12 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Le istruzioni

- Le istruzioni C terminano con un punto e virgola (;)
- Esempio: `printf("Stampa questa riga \n");`
- Generalmente, un'istruzione può essere interrotta e ripresa nella riga successiva, dal momento che la sua conclusione è dichiarata chiaramente dal punto e virgola finale.
- Raggruppamenti di istruzioni si fanno utilizzando le parentesi graffe { }
- Esempio: `<istruzione>; { <istruzione>; <istruzione>; }`

back X next

Federica 27/09/2007 13 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Struttura del Sorgente (2)

- Altri oggetti fondamentali di un codice sorgente in linguaggio C sono:
 - **Direttive del preprocessore**
 - **Commenti**
- Le direttive del preprocessore guidano alla compilazione del codice. L'uso più comune riguarda l'inclusione di codice esterno al file sorgente (librerie), composto da file che terminano con l'estensione ".h". Tali istruzioni iniziano con il simbolo "#".
- I commenti vengono indicati tra i simboli /* e */.

back X next

Federica 27/09/2007 14 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Esempio di Programma C

- Il seguente è un semplice esempio di programma scritto in linguaggio C


```
#include <stdio.h>
main() /* esempio di programma C */
{
printf("Primo programma C\n");
}
```
- Questo programma ha una sola funzione "**main()**".
- In un programma C esiste un solo **main** e l'esecuzione del programma corrisponde alla chiamata di tale funzione.
- Si notino inoltre la parentesi graffa aperta "{ " per l'inizio del corpo della funzione e la parentesi graffa chiusa "}" per la fine del corpo della funzione

back X next

27/09/2007

15

Facoltà di Scienze
Matematiche
Fisiche Naturali

Introduzione al C

Standard C library

- Esaminiamo l'istruzione

```
printf ("Primo programma C\n");
```
- **printf** è il nome di una funzione il cui codice è già scritto ed inserito nella **standard C library "stdio.h"**.
- Questa libreria viene inclusa durante la fase di compilazione del programma attraverso la direttiva del preprocessore **#include <stdio.h>** .
- La libreria standard è necessaria alla gestione dei flussi di standard input, standard output e standard error.
- Altre funzioni appartenenti a questa libreria sono sono:
 - **printf()**, **fprintf()**, **sprintf()**, **scanf()**, **fscanf()**, **sscanf()**, **getc()**, **gets()**, **getchar()**, **putc()**, **puts()**, **putchar()**, **fgetc()**, **fgets()**, **fputs()**, **fwrite()**, **fread()**, etc...

back ✖ next

27/09/2007

16

Facoltà di Scienze
Matematiche
Fisiche Naturali

Introduzione al C

Le variabili

- Una **variabile** è un'astrazione di una o più celle di **memoria**.
- Un **identificatore** è il nome associato ad un variabile e consiste di un numero qualsiasi di caratteri alfanumerici minuscoli o maiuscoli incluso il carattere "_" (underscore).
- Il primo carattere deve essere una lettera oppure underscore.
- Il compilatore fa differenza tra lettere minuscole e maiuscole.
- **Esempi validi:** `sp_addr` `sp2_addr` `F_lock_user_found`
- **Esempi non validi:** `20_secolo` `-pippo`
- Ogni identificatore possiede due attributi che lo caratterizzano:
 - **classe di memoria:** determina la durata della memoria associata alla variabile
 - **Tipo:** determina il significato dei valori assunti dalla variabile

back ✖ next

Federica 27/09/2007 17

Facoltà di Scienze
Matematiche
Fisiche Naturali

Introduzione al C

Parole chiave

- Alcuni identificatori sono parole riservate (**keywords**) del linguaggio e pertanto non possono essere usate come nomi di variabili.
- Le **keywords** del linguaggio C standard sono 32:

<code>auto</code>	<code>double</code>	<code>int</code>	<code>struct</code>
<code>break</code>	<code>else</code>	<code>long</code>	<code>switch</code>
<code>case</code>	<code>enum</code>	<code>register</code>	<code>typedef</code>
<code>char</code>	<code>extern</code>	<code>return</code>	<code>union</code>
<code>const</code>	<code>float</code>	<code>short</code>	<code>unsigned</code>
<code>continue</code>	<code>for</code>	<code>signed</code>	<code>void</code>
<code>default</code>	<code>goto</code>	<code>sizeof</code>	<code>volatile</code>
<code>do</code>	<code>if</code>	<code>static</code>	<code>while</code>

back X next

Federica 27/09/2007 18

Facoltà di Scienze
Matematiche
Fisiche Naturali

Classi di Memoria

Classi di Memoria

- Esistono due classi di memoria: **automatica** e **statica**
- La classe di memoria automatica è relativa a quegli oggetti locali ad un blocco (funzione o programma) che viene liberata non appena si raggiunge la fine di quel blocco.
- La classe di memoria statica è relativa a quegli oggetti locali ad un blocco od esterni a qualsiasi blocco che non viene liberata tra uscite ed entrate successive tra diversi blocchi.
- Se non esistono altre specificazioni ogni oggetto dichiarato in un blocco ha classe di memoria automatica, se la loro definizione è accompagnata dalla parola chiave "**static**", allora la loro classe di memoria è statica.
- Gli oggetti dichiarati all'esterno di qualsiasi blocco (variabili globali) hanno sempre classe di memoria statica.

back X next

Federica 27/09/2007 19 Facoltà di Scienze Matematiche Fisiche Naturali

Tipi di dati

Tipi di dati

- Tutte le variabili devono essere dichiarate con il loro tipo prima di essere utilizzate
- Ci sono 4 tipi base in C:
 - **char** (carattere): 8 bit
 - **int** (intero): 16 bit
 - **float** (numero in virgola mobile): 32 bit
 - **double** (float a doppia precisione): 64 bit

back X next

Federica 27/09/2007 20 Facoltà di Scienze Matematiche Fisiche Naturali

Interi

Interi

- La dichiarazione di variabili intere deve essere posta all'inizio di un blocco di codice.
- **Esempio: `int a,b,c;`**
- Dopo che una variabile è stata dichiarata è possibile assegnarle un valore intero tramite l'operatore di assegnamento
- **Esempio: `a = 100;`**
- Avendo a disposizione un calcolatore che memorizza interi a 16 bit, i due valori estremi che possono essere assegnati ad un intero sono: -32768 e +32767

back X next

Federica 27/09/2007 21 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Caratteri

- Le variabili di tipo carattere vengono dichiarate nel modo seguente:
`char anno, mese;`
- Per assegnare un valore carattere **A** ad una variabile **c** di tipo **char** usiamo la seguente sintassi
`c='A'`
- In effetti, nel linguaggio C i caratteri sono valori numerici ai quali per convenzione sono associate lettere dell'alfabeto, segni di interpunzione ed altri simboli alfanumerici.
- La codifica più comunemente usata è la codifica **ASCII** (American Standard Code for Information Interchange)
- Ad esempio la lettera 'A' viene codificata come 65 mentre la minuscola 'a' corrisponde a '97'

back ✖ next

Federica 27/09/2007 22 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Dati in virgola mobile

- I dati floating point sono una approssimazione dei reali, espressi come frazioni decimali.
- Per esempio: 3.14159, 2.71828
- Solitamente vengono utilizzati 32 bit per la rappresentazione interna di questi dati, di cui almeno i primi 6 sono i più significativi.
- Il range di variazione dei floating point è: 1...e-39 fino a 1...e+38

back ✖ next

Federica 27/09/2007 23 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Costanti

- In C esistono diversi tipi di costanti, in cui possiamo distinguere 5 tipi fondamentali:
 - costanti **intere**
 - costanti esplicitamente **long**
 - costanti **carattere**
 - costanti in **virgola mobile**
 - **stringhe di caratteri** costanti

Costanti particolari:

- New line (lf) `'\n'`
- Carriage return (cr) `'\r'`
- Backspace (bs) `'\b'`
- Horizontal tab (tab) `'\t'`
- Backslash (\) `'\\'`

back X next

Federica 27/09/2007 24 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Operatori

- Gli operatori sono classificati secondo le tre seguenti categorie

<h4>Aritmetici:</h4> <ul style="list-style-type: none"> ▪ <code>"+"</code> somma, ▪ <code>"-"</code> sottrazione, ▪ <code>"*"</code> moltiplicazione, ▪ <code>"/"</code> divisione, ▪ <code>"%"</code> modulo 	<h4>Relazionali:</h4> <ul style="list-style-type: none"> ▪ <code>"<"</code> minore di, ▪ <code>">"</code> maggiore di, ▪ <code>"<="</code> minore o uguale, ▪ <code>">="</code> maggiore o uguale, ▪ <code>"=="</code> uguale a, ▪ <code>"!="</code> diverso da
<h4>Logici:</h4> <ul style="list-style-type: none"> ▪ <code>"&&"</code> AND logico <code>"&"</code> AND bit a bit ▪ <code>" "</code> OR logico <code>" "</code> OR bit a bit ▪ <code>"!"</code> NOT logico <code>"~"</code> NOT bit a bit 	

back X next

Federica 27/09/2007 25 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Operatori aritmetici e di assegnamento

- Si consideri l'espressione:
`anno = anno + 1;`
 che significa: "prendere il valore corrente della variabile anno, sommargli 1 e memorizzare il risultato nuovamente in anno".
- Nell'espressione sono presenti due operatori "+" e "=" che implicano l'esecuzione dell'operazione in due passi.
- L'ordine di esecuzione è: somma e poi assegnamento.

back X next

Federica 27/09/2007 26 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Operatori di incremento e decremento

- Questi operatori sommano o sottraggono 1 all'operando cui sono applicati.
- Esempi: `a++` incrementa `a` di 1; `a--` decrementa `a` di 1
- La posizione degli operatori di incremento e decremento può essere *prefissa* o *suffissa*.
- Esempio: (per `a=5`):**
 - `b = ++a;` → `a=6` e `b=6`
 - `b = a++;` → `a=6` e `b=5`

Operatori di assegnamento composti

- Sono operatori assegnamento opportunamente composti con quelli aritmetici
- L'elenco completo degli assegnamenti è il seguente:

`=` `+=` `-=` `*=` `/=` `%=`

back X next

Federica 27/09/2007 27 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Printf e Scanf con formattazione

<ul style="list-style-type: none"> • Si consideri il seguente codice: <pre style="margin-left: 20px;"> #include <stdio.h> main() { int a,b,c; printf("\nIl primo numero e'"); scanf("%d",&a); printf("\nIl secondo numero e'"); scanf("%d",&b); c=a+b; printf("\nIl totale e' %d \n",c); } </pre>	<ul style="list-style-type: none"> • Alcuni tipi di % che possono essere usati in ANSI C sono: <ul style="list-style-type: none"> ▪ %c char single character ▪ %d (%i) int signed integer ▪ %e float or double exponential format ▪ %f float or double signed decimal ▪ %s array of char sequence of characters
--	--

back ✖ next

Federica 27/09/2007 28 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Strutture di controllo

- **SEQUENZIALI**
 - Statement semplice
 - Statement composto
- **CONDIZIONALI**
 - if (< expr.>) { } else { }
 - switch(<expr.>) { case < cost> : .. }
- **ITERATIVE**
 - while(<expr.>) { }
 - for (..) { }
 - do { } while(< expr.>)

back ✖ next

Federica 27/09/2007 29 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Statement Semplici

- Uno statement semplice può essere :
 - assegnamento
 - espressione
 - chiamata a funzione
- Esempio:


```
main()
{
int x;
x = - 456; /* assegnamento */
x = x + 1; /* espressione */
printf (" X = %d \n ", x); /* funzione */
}
```

back ✖ next

Federica 27/09/2007 30 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Istruzione "IF - ELSE"

- Lo statement "if - else" è usato per prendere delle decisioni all'interno di un programma
- Sintassi:


```
if ( espressione )
istruzione;
```
- Sintassi alternativa:


```
if ( espressione ) {
istruzione_1;
istruzione_2; }
else
istruzione_3;
```
- Semantica
 - L'espressione viene valutata e, se risulta essere TRUE o NON ZERO, viene eseguito il set di istruzioni corrispondente.

Esempio:

```
if ( x > 0 )
{
        if ( k == m )
            y = m;
}
else
{
        y = 0;
        k = 100;
}
```

back ✖ next

Federica 27/09/2007 31 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Operatore Ternario

- L'operatore ternario è una forma compressa dell'operatore **if - else**.
- **Sintassi:**
`< expr > ? < expr1 > : < expr2 >;`
- **Semantica:**
 - Se `< expr >` è vero viene valutato solo `< expr1 >` altrimenti viene valutato `< expr2 >`
- Esempio:
`max = (alfa > beta) ? alfa : beta;`

che corrisponde a...

```
if (alfa > beta)
    max = alfa;
else
    max = beta;
```

back X next

Federica 27/09/2007 32 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Istruzione else-if

- Quando occorre effettuare una scelta plurima è possibile, ma non consigliabile, utilizzare l'istruzione else-if.
- **Sintassi:**

```
if (espressione)
    <istruzione/i>
else if (espressione)
    <istruzione/i>
else if (espressione)
    <istruzione/i>
else
    <istruzione/i>
```

back X next

Federica 27/09/2007 33 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Istruzione Switch

- Trasferisce l'elaborazione a uno o più statement composti, in funzione della valutazione di una espressione.
- È usata quando occorre effettuare scelte plurime.
- L'esecuzione delle istruzioni avviene in modo SEQUENZIALE a partire dalla prima espressione `cost_x` che risulta vera e fino al termine dell'istruzione switch.
- Ciò permette di associare più possibilità ad un'unica etichetta.
- Per evitare l'esecuzione di istruzioni successive a quelle corrispondenti alla condizione soddisfatta si usa il comando `break`

Sintassi:

```
switch ( <expr> )
{
case <cost_1>:
    statement;
case <cost_2>:
    statement;
case <cost_3>:
    statement;
default
    statement;
}
```

back ✖ next

Federica 27/09/2007 34 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Esempio di Switch

```
#include <stdio.h>
main()
{
    int i;
    printf("Enter a number between 1 and 3");
    scanf("%d",&i);
    switch (i)
    {
        case 1:
            printf("one"); break;
        case 2:
            printf("two"); break;
        case 3:
            printf("three"); break;
        default:
            printf("unrecognized number");
    } /* end of switch */
}
```

back ✖ next

Federica 27/09/2007 35 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

<h3>Istruzione "while"</h3> <ul style="list-style-type: none"> ▪ L'istruzione while permette di eseguire una serie di istruzioni fintanto che una condizione iniziale rimane TRUE. ▪ Sintassi: <pre>while (<expr>) { statement; }</pre> ▪ Esempio <pre>a = 1; while (a <= 100) { total += a*a; a += 1; }</pre> 	<h3>Istruzione "do-while"</h3> <ul style="list-style-type: none"> ▪ L'istruzione do while permette di eseguire delle istruzioni e poi di rieseguirle fintanto che una condizione iniziale rimane TRUE. ▪ Dunque, un ciclo do while è eseguito almeno una volta e la condizione viene valutata dopo l'esecuzione delle istruzioni. ▪ Sintassi: <pre>do { statement; } while (<expr>);</pre>
--	---

back ✖ next

Federica 27/09/2007 36 Facoltà di Scienze Matematiche Fisiche Naturali

Introduzione al C

Istruzione "for"

- L'istruzione for è molto simile al while.
- **Sintassi:**

```
for ( <inizializza var>; <expr>; <aggiorna var> ) {
    statement;
}
```
- Equivalente alla struttura seguente che utilizza while:

```
< inizializza variabili >;
while ( <expr> ) {
    < statement >;
    < aggiorna variabili >;
}
```
- **Esempio:**

```
main()
int sum , i;
sum = 0;
for ( i = 1 ; i <= 10; i++ ) {
    sum += i;
}
```

back ✖ next

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.