




**UNIVERSITÀ DEGLI STUDI  
DEL SANNIO** Benevento  
**DING**  
DIPARTIMENTO DI INGEGNERIA

**CORSO DI "PROGRAMMAZIONE I"**

Prof. Franco FRATTOLILLO  
Dipartimento di Ingegneria  
Università degli Studi del Sannio

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio      Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT      1




**Esercizio**

- Determinare il valore di x e y

```
#include <stdio.h>
main() {
    int x, y;
    x=2; y=4;
    y=3*y;
    x=y+x;
    printf("%d\n", x);
    printf("%d\n", y);
}
```

memoria	
x	14
y	12

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio      Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT      2




**Esercizio**

- Determinare i valori di p, q ed r al termine delle seguenti istruzioni:

```
p=2; q=3; r=q;
q=q+1;
p=p+q-(2*r);
r=p+r+1;
```

memoria	
p	0
q	4
r	4

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio      Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT      3



**Esercizio**

- Come si fa a scambiare il valore di due variabili?


• Prima

x	3
y	5

• Dopo

x	5
y	3

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio      Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT      4



Esercizio: scambio variabili

• È corretto questo programma?

```
#include <stdio.h>
main( ) {
  int x, y;
  x=3; y=5;
  x=y; y=x;
  printf("%d\n", x);
  printf("%d\n", y);
}
```


**NO!**

alla fine ottengo:

**x** vale 5

**y** vale 5

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
5




Soluzione

```
#include <stdio.h>

main( ) {
  int x, y, temp;
  x=3; y=5;
  temp=x;
  x=y;
  y=temp;
  printf("%d\n", x);
  printf("%d\n", y);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
6




È corretto questo programma?

```
#include <stdio.h>
main( ){
  int y;
  y=0;
  x=y+12;
  printf("%d\n", x);
  printf("%d\n", y);
}
```

**NO!!**

**uso x senza averla dichiarata**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
7



L'input

• ... leggere l'input??

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
8

## Area del rettangolo

- L'utente deve poter immettere i valori di base e altezza

```
/* Calcolo area rettangolo */
#include <stdio.h>
main() {
    int base, altezza, area;
    printf("Valore base: ");
    scanf("%d", &base);
    printf("Valore altezza: ");
    scanf("%d", &altezza);
    area = base*altezza;
    printf("Area: %d", area);
}
```

## Input

- `scanf("%d", &base);`

Leggi un intero e assegnalo alla variabile **base**

valore intero in formato decimale, che verrà assegnato alla variabile **base**

indirizzo di memoria in cui si trova la variabile **base**

si scrive `&base`


## Risultato

```
admin@VAIO-
$ gcc rettangolo.c -o rettangolo.exe

admin@VAIO-
$ ./rettangolo.exe
Valore base: 5
Valore altezza: 7
Area: 35
admin@VAIO-
$ |
```

## Errori tipici


- Dimenticare una o entrambe le virgolette che circondano la stringa di controllo del formato in una *printf* o in una *scanf*
- Dimenticare il % in una *printf* o in una *scanf*
- Dimenticare il simbolo & in una *scanf*



## ... abbiamo visto

- Struttura di un programma
- Commenti
- Dichiarazioni di variabili
- Istruzioni semplici:
  - istruzione di uscita
  - istruzione di ingresso
  - istruzione di assegnamento

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    13




## Riepilogo: struttura di un programma

direttive

```
main( ) {
    dichiarazioni;
    istruzioni;
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    14



## Riepilogo: commenti

- // commenti  
in questo modo il compilatore non valuta una riga intera
- /\* commenti \*/  
in questo modo il compilatore non valuta quello che c'è nel mezzo

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    15



## Riepilogo: dichiarazioni di variabili

```
int x, y, z;
```

- Dichiarazione di tre variabili (x, y, z) di tipo intero

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    16

## ... ancora sulle dichiarazioni

- Nelle dichiarazioni può essere anche definito un valore iniziale, che viene automaticamente assegnato alle posizioni di memoria

- Esempio:

```
int y;  
int x=5;
```

## Identificatori ...


- Un identificatore non è altro che il nome associato ad un oggetto quale può essere una variabile o una funzione
- Ogni identificatore, quando usato per riferire una variabile, è caratterizzato da due attributi:
  - classe di memoria
  - tipo
- La "classe di memoria" determina il tempo di vita dell'identificatore
- Il "tipo" determina il campo dei valori che può assumere l'identificatore e le operazioni che sono definite su di esso

## Identificatori ...

- Gli identificatori di qualsiasi oggetto in un programma possono essere specificati da insiemi di caratteri alfanumerici minuscoli o maiuscoli incluso il carattere "\_".
  - il primo carattere deve essere una lettera oppure il carattere "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

## Circa gli identificatori ...


- Utilizzate identificatori significativi
  - possibilmente consistenti con il vocabolario del dominio d'interesse
- Evitare:
  - nome del proprio cane, fidanzato/a etc.: bob, jane, pluto
  - stati d'animo: uffa ...
  - sequenze casuali di caratteri: qwewn, wqeu90qw8u
  - singole lettere (ok per gli indici, meno bene per altre variabili ...)
- Possiamo scrivere identificatori lunghi...(non molto ...)
  - parole composte usando "camel case" o underscore
    - checkDate o check\_date
- Abbreviazioni e acronimi ok, ma facendo attenzione
  - "ctr" significa "control" o "counter" ?



## Riepilogo: istruzione di uscita (output)

- `printf("stringa")`
- stampa tutti i caratteri che compongono stringa, uno dopo l'altro
- Nota: stringa è semplicemente una sequenza di caratteri


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    21



## Riepilogo: istruzione di uscita (output)

- `printf`: usata anche per stampare il contenuto di una variabile
- Sintassi:  
`printf(< stringa di formato >, < argomenti >)`
- < stringa di formato >: messaggio che deve essere visualizzato, comprensivo dei riferimenti ai tipi di dati contenuti nelle variabili
- < argomenti >: variabili contenenti i dati da visualizzare

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    22




## Riepilogo: istruzione di uscita (output)

- Bisogna specificare il formato della variabile utilizzando il carattere speciale di formattazione % seguito dal carattere che definisce un certo formato per una variabile: %d è usato per il tipo intero

```
printf("%d %d %d", x, y, z);
```

- Provoca la stampa di tre valori interi contenuti in x, y e z
- Ogni occorrenza del carattere % nel primo argomento è associata al corrispondente argomento di `printf`, a partire dal secondo: affinché non si verifichino errori, la corrispondenza deve riguardare sia il numero che il tipo di argomenti

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    23




## Esempi di printf

```
#include<stdio.h>
main( ) {
    int x=3;
    int y=5;
    printf("Valore di x: %d\n", x);
    printf("Valore di y: %d\n", y);
}
```

**Valore di x: 3**  
**Valore di y: 5**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    24




## Cosa stampa?

```
int a=0;
int b=2;
int c=4;
int d=6;
int e=8;
int f=10;
printf("Numeri pari da 1 a 10: %d %d %d %d %d\n",a,b,c,d,e);
```

Numeri pari da 1 a 10: 0 2 4 6 8

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    25




## Cosa stampa?

```
int a=0;
int b=2;
int c=4;
int d=6;
int e=8;
int f=10;
printf("Numeri pari da 1 a 10: %d %d %d %d %d %d\n",a,b,c,d,e,f);
```

Numeri pari da 1 a 10: 0 2 4 6 8 10

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    26



## Ancora sulla printf


- I calcoli possono essere eseguiti all'interno di una printf

```
#include<stdio.h>
main() {
    int x=3;
    int y=5;
    x=y+x;
    printf("x=%d\n", x);
}
```

```
#include<stdio.h>
main() {
    int x=3;
    int y=5;
    printf("x=%d\n", y+x);
}
```

Entrambi stampano: **x=8**


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    27



## Riepilogo: istruzione di ingresso (input)

- scanf: usata per effettuare la lettura di una variabile
- Sintassi:  
scanf(< stringa di formato >, < argomenti >)
- < stringa di formato >: indica il formato in cui saranno inseriti i valori
- < argomenti >: variabili che devono contenere i valori inseriti, preceduti dal carattere &


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    28



### Riepilogo: istruzione di ingresso (input)

- `scanf("%d", &x)`
- legge la prossima sequenza di caratteri sullo standard input che rappresentano un valore di tipo intero e assegna tale valore alla variabile `x`;
- occorre premettere al nome della variabile il carattere `&`
- Posso anche scrivere:  
`scanf("%d %d", &x, &y);`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    29




### Riepilogo: istruzione di assegnamento

`x = 1 + 4 * (y + 2);`

- calcola il valore dell'espressione a destra del segno `=` e lo assegna alla variabile `x`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    30



### Esercizio

- Scrivere un programma che, utilizzando una sola istruzione `printf`, visualizzi:


Prove

tecniche

di

visualizzazione

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    31



### Soluzione

```
#include<stdio.h>
main( ) {
    printf("Prove\n\n tecniche\ndi\n visualizzazione");
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    32



## Esercizio

- Che sarà visualizzato (se lo sarà), quando ognuna delle seguenti istruzioni C verrà eseguita? Assumere che  $x=2$  e  $y=3$

```
printf("%d", x);
printf("%d", x+x);
printf( «x=%d», x);
z=x+y;
scanf("%d %d", &x, &y);
/* printf("x+y=%d", x+y)*/;
```

**Il programma  
non compila**

## Esercizio

- Scrivere un programma che richieda all'utente un numero intero. Il programma deve quindi calcolare e visualizzare il doppio

**Esempio:**

**Inserisci un numero**

**5**

**Il risultato = 10**

## Soluzione

```
#include<stdio.h>
main( ) {
    int numero;
    printf("Inserisci un numero\n");
    scanf("%d", &numero);
    numero = numero+numero;
    printf("Il risultato = %d", numero);
}
```

## Compilazione ed esecuzione



```
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$ gcc esempio.c -o esempio.exe
admin@VAIO-
$ ./esempio.exe
Dammi un numero
5
Il risultato = 10
admin@VAIO-
$ ./esempio.exe
Dammi un numero
7
Il risultato = 14
admin@VAIO-
$
```

## Esercizio

- Scrivere un programma che richieda all'utente tre numeri interi. Il programma deve quindi calcolare la differenza dei tre numeri e moltiplicare il risultato per il primo numero. Visualizzare il risultato

Esempio:

**Inserisci il primo numero 10**

**Inserisci il secondo numero 2**

**Inserisci il terzo numero 3**

**Il risultato = 50**

## Soluzione

```
#include<stdio.h>
main() {
    int primo, secondo, terzo, risultato;
    printf("Inserisci il primo numero ");
    scanf("%d", &primo);
    printf("Inserisci il secondo numero ");
    scanf("%d", &secondo);
    printf("Inserisci il terzo numero ");
    scanf("%d", &terzo);
    risultato = (primo-secondo-terzo)*primo;
    printf("Il risultato = %d", risultato);
}
```

## Compilazione ed esecuzione



```
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$ gcc esempio.c -o esempio.exe
admin@VAIO-
$ ./esempio.exe
Dammi il primo numero 10
Dammi il secondo numero 3
Dammi il terzo numero 2
Il risultato = 50
admin@VAIO-
$ ^C
admin@VAIO-
$
```

## Soluzione alternativa

```
#include<stdio.h>
main() {
    int primo, secondo, terzo, risultato;
    printf("Inserisci i tre numeri separati da spazio\n");
    scanf("%d %d %d", &primo, &secondo, &terzo);
    risultato = (primo-secondo-terzo)*primo;
    printf("Il risultato = %d", risultato);
}
```

## Compilazione ed esecuzione

```

$
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$
admin@VAIO-
$ gcc esempio.c -o esempio.exe
admin@VAIO-
$ ./esempio.exe
Dammi i tre numeri separati da spazio
10 2 3
Il risultato = 50
admin@VAIO-
$

```

## Esercizio

- Indicare tutti gli errori commessi nel seguente programma e fornire una versione corretta

```

#include<stdio.h>;
/* programma con errori */
main( )
Printf("programmi con errori");
int num = 9
scanf("d", num);
ris+2 = num;
printf("Risultato=" ris);
}

```

## Versione corretta

```

#include<stdio.h>
/* programma con errori */
main( ) {
int ris;
int num;
printf("programmi con errori");
scanf("%d", &num);
ris = num-2;
printf("Risultato=%d", ris);
}

```

## Esercizio

- Cosa stampa il seguente programma?

```

#include<stdio.h>
main( ) {
int x=2, y=3;
scanf("%d %d", &x, &y);
y=x*y;
printf("Il prodotto x*y = y");
}

```

### Soluzione

- Stampa:  
Il prodotto  $x*y = y$
- Se avessi voluto stampare il prodotto di due numeri letti dall'esterno, avrei dovuto usare questa istruzione:
- `printf("Il prodotto  $x*y = %d$ ", y);`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 45

### Tipi di dati

- Definiscono i VALORI che una variabile può assumere
- Definiscono le OPERAZIONI che possono essere effettuate su tali variabili

Tipi

```

graph TD
    Tipi --> Semplici
    Tipi --> Strutturati
    Semplici --> intero
    Semplici --> reale
    Semplici --> carattere
  
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 46

### Tipo intero

- Il tipo intero viene utilizzato per tutte le grandezze che possono essere rappresentate come numeri interi, come per esempio, età, numero di figli, ecc

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 47

### Campo di variabilità

- Intervallo finito

Tipo	Dimensione (byte)	Valore minimo	Valore massimo
short int	2	-32768	+32767
int	4	$-2^{31}$	$2^{31} - 1$
long int	8	$-2^{63}$	$2^{63} - 1$

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 48

## Notazione per i valori costanti

- Sequenza di cifre, preceduta eventualmente dal segno + o -

```
int x;

x = 356;
x = -987;
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    49

## Metodi di input/output

```
int x;
```

- Output  
`printf("%d", x);`
- Input  
`scanf ("%d", &x);`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    50

## Operazioni

- + somma
- meno unario
- differenza
- \* prodotto
- / divisione intera
- % resto della divisione intera (modulo)

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    51

## Esempi

```
int x=9;
int y=6;
```

<b>x/y</b>		<b>1</b>
<b>x%y</b>		<b>3</b>

- Se  $y=0 \rightarrow$  non è possibile eseguire una divisione per zero
- Per i numeri negativi, la direzione di troncamento del / ed il segno del risultato di % dipendono dalla macchina
- Se  $x\%y$  restituisce 0  $\rightarrow$  x è multiplo di y

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    52

### Dettagli sul tipo intero

- Possibile aggiungere il qualificatore "unsigned" alla definizione di tipo, che consente alla variabile di assumere solo numeri positivi

Tipo	Dimensione (byte)	Valore minimo	Valore massimo
<b>unsigned short int</b>	<b>2</b>	<b>0</b>	<b>65535</b>
<b>unsigned int</b>	<b>4</b>	<b>0</b>	<b><math>+2^{32} - 1</math></b>
<b>unsigned long int</b>	<b>8</b>	<b>0</b>	<b><math>+2^{64} - 1</math></b>

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 53

### short e long

- Sono qualificatori e si possono applicare agli interi
  - short utilizza un numero di bit ridotto per rappresentare gli interi
    - int a; /\* intero 32 bit\*/
    - short int a; /\* intero 16 bit \*/
    - short a; /\* int può essere omissso \*/
  - long utilizza un maggior numero di bit per rappresentare gli interi
    - long int a; /\* intero a 64 bit \*/
    - long a; /\* int può essere omissso \*/

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 54

### Costanti intere

- Costanti intere decimali: 0, 1, 8, 10, 256
- Costanti intere ottali: 0, 01, 010, 012, 0377
- Costanti intere esadecimali (precedute da 0x): 0x0, 0x1, 0xA, 0xC, 0xFF
- Esempio:
 

```
main() {
    int a;
    a = 0xA;
    a = 10;
    a = 012;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 55

### Costanti long o unsigned

- Una costante intera seguita da una L è di tipo long, mentre seguita da una U è di tipo unsigned:
  - 0L, 1L, 8L, 10L, 256UL
- Esempio:
 

```
main() {
    long numero;
    numero = 0L;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 56

## Precedenze

- Il C dispone di un insieme di regole che determinano l'ordine in cui le varie operazioni devono essere eseguite
- Per le operazioni aritmetiche le precedenze sono quelle definite in matematica

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    57

## Precedenza degli operatori aritmetici

Operatori	Associatività
Parentesi: ()	dall'interno all'esterno
Operatore unario: -	da destra a sinistra
Operatori binari: * / %	da sinistra a destra
Operatori binari: + -	da sinistra a destra

↑ + alta  
↓ + bassa

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    58

## Associatività

$2*3*5 \quad \rightarrow \quad ((2*3)*5)$


- da sinistra a destra

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    59

## Regole

- In base alla precedenza
- Se tutti gli operandi hanno la stessa precedenza: associatività
- La priorità degli operatori può essere alterata con le parentesi tonde: vengono valutate per prima le operazioni all'interno delle parentesi tonde più interne


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    60



Esempio

• $3*5+2$	⇒	17 * precedenza su +
• $2*3/2*3$	⇒	9 * e / stessa precedenza: sx verso dx: $((2*3)/2)*3$
• $3*(5+2)$	⇒	21
• $7+3-15+4*5$	⇒	15 $(( (7+3) - 15 ) + (4*5) )$


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
61



Esercizio 1

- Scrivere un programma che richiede all'utente un numero che rappresenta un'altezza in centimetri. Il programma converte tale altezza in metri e centimetri e visualizza il risultato in metri e centimetri
- Esempio:
  - Utente immette 134cm → 1 m, 34 cm
  - Utente immette 45cm → 0 m, 45 cm
  - Utente immette 200cm → 2 m, 0 cm


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
62



Soluzione: esercizio 1

```
#include<stdio.h>
void main( ) {
    int numero, metri, centimetri;
    printf("Inserisci l'altezza in cm: ");
    scanf("%d", &numero);
    metri = numero/100;
    centimetri= numero%100;
    printf("%d m, %d cm", metri, centimetri);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
63



Esercizio 2

- Scrivere un programma che richiede all'utente un numero positivo di tre cifre. Il programma stampa la cifra centrale
- Esempio:
  - 356 → 5
  - 789 → 8

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
64



## Soluzione: esercizio 2

```
#include<stdio.h>
main( ) {
    int n, cfr;
    printf("Inserisci un numero positivo di tre cifre: ");
    scanf("%d", &n);
    cfr = (n/10) % 10;
    printf("Cifra centrale: %d", cfr);
}
```

## Esercizio 3


- Rovesciare un numero positivo di tre cifre
- Esempio:
  - 356 diventa 653
  - 789 diventa 987

## Soluzione: esercizio 3

```
#include<stdio.h>
main( ) {
    int n, unita, decine, centinaia;
    printf("Inserisci un numero positivo di tre cifre: ");
    scanf("%d", &n);
    unita = n % 10;
    decine = (n/10) % 10;
    centinaia = n/100;
    printf("Numero rovesciato: %d%d%d", unita, decine,
        centinaia);
}
```

## ... oppure


```
#include<stdio.h>
main( ) {
    int n, unita, decine, centinaia;
    printf("Inserisci un numero positivo di tre cifre: ");
    scanf("%d", &n);
    unita = n % 10;
    decine = (n/10) % 10;
    centinaia = n/100;
    printf("NumRov: %d", 100*unita+10*decine+centinaia);
}
```



## Esercizio

- Scrivere un programma che richiede all'utente un numero che rappresenta un periodo di tempo espresso in minuti. Il programma converte tale periodo in ore e minuti e visualizza il risultato in ore e minuti
- Esempio
  - Utente immette 134m → 2 h, 14 m
  - Utente immette 45m → 0 h, 45 m
  - Utente immette 180m → 3 h, 0 m

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
69




## Soluzione

```

#include<stdio.h>
main( ) {
    int numero, minuti, ore;
    printf("Inserisci il tempo in minuti: ");
    scanf("%d", &numero);
    ore = numero/60;
    minuti= numero%60;
    printf("%d h, %d m", ore, minuti);
}
  
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
70



## Tipo reale

- I numeri reali vengono usati per rappresentare prezzi, pesi, misure, o per calcoli matematici, ecc.


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
71



## Campo di variabilità

Tipo	Dimensione (byte)	Valore minimo	Valore massimo
<b>float</b> precisione singola	<b>4</b>	<b>-3.2·10<sup>±38</sup></b>	<b>+3.2·10<sup>±38</sup></b>
<b>double</b> precisione doppia	<b>8</b>	<b>-1.7·10<sup>±308</sup></b>	<b>1.7·10<sup>±308</sup></b>

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
72



## Notazione per i valori costanti

- Esistono due modi di scrivere numeri reali:
- parte intera, punto, parte decimale  
4.34
- parte intera, e o E, esponente con segno  
-3E3 rappresenta  $-3 \cdot 10^3$  cioè -3000  
5e-2 rappresenta  $5 \cdot 10^{-2}$  cioè 0.05


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    73



## Operazioni

- somma, differenza unaria e binaria, prodotto,
- divisione reale, esponenziali, logaritmi, funzioni trigonometriche, ...


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    74



## Funzioni aritmetiche

- `#include<math.h>`
- x e y di tipo double e restituiscono un double
- `pow(x, y)`     $x^y$
- `sin(x)`    seno di x, con x espresso in radianti
- `cos(x)`    coseno di x, con x espresso in radianti
- `exp(x)`     $e^x$
- `log(x)`    logaritmo naturale di x
- `sqrt(x)`    radice quadrata x,  $x \geq 0$
- `log10(x)`    logaritmo in base 10 di x
- .....

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    75




## Metodi di input/output

`float x;`

- Output  
`printf("%f", x);`
- Input  
`scanf ("%f", &x);`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    76




## Metodi di input/output

```
double x;
```

- Output  
`printf("%f", x);`
- Input  
`scanf ("%f", &x);`


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 77



## Esercizio 1

- Scrivere un programma che calcola l'area di un cerchio di raggio  $r$  immesso dall'utente


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 78



## Soluzione: esercizio 1

```
#include<stdio.h>
main( ) {
    float r, area;
    printf("Inserisci il raggio: ");
    scanf("%f", &r);
    area = r*r*3.14;
    printf("L'area del cerchio di raggio %f è uguale a
           %f\n", r, area);
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 79



## Esercizio 2

- Scrivere un programma che effettua la conversione da LIRE ITALIANE a EURO
- Esempio:
  - se immetto 1000 LIRE
  - 1000 LIRE = 0.516457 EURO


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 80



## Soluzione: esercizio 2

```
#include<stdio.h>
main( ) {
    float euro, lira;
    printf("Inserisci il numero in LIRE: ");
    scanf("%f", &lira);
    euro = lira / 1936.27;
    printf("%f LIRE = %f EURO", lira, euro);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    81




## Migliorare stampa

```
#include<stdio.h>
main( ) {
    float euro, lira;
    printf("Inserisci il numero in LIRE: ");
    scanf("%f", &lira);
    euro = lira / 1936.27;
    printf("%.0f LIRE = %.2f EURO", lira, euro);
}
```

**%.nf**  
 stampa un numero  
 frazionario con **n**  
 cifre dopo il punto  
 decimale


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    82



## Esercizio

- Scrivere un programma che calcola l'area di un triangolo di base  $b$  ed altezza  $h$ , immessi dall'utente


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    83



## Soluzione

```
#include<stdio.h>
main( ) {
    float base, altezza, area;
    printf("Inserisci la base: ");
    scanf("%f", &base);
    printf("Inserisci l'altezza: ");
    scanf("%f", &altezza);
    area = (base * altezza)/2;
    printf("Area = %f\n", area);
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    84



Esercizio

- Scrivere un programma che effettua la conversione da EURO a LIRE ITALIANE
- Esempio:
  - se immetto 100 EURO
  - 100 Euro = 193627 LIRE

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
85




Soluzione

```

#include<stdio.h>
main( ) {
    float euro, lira;
    printf("Inserisci il numero in EURO: ");
    scanf("%f", &euro);
    lira = euro * 1936.27;
    printf("%f EURO = %f LIRE", euro, lira);
}
  
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
86



Tipo carattere

- Finora abbiamo lavorato con valori numerici
- I numeri costituiscono molta parte del lavoro dei computer, ma non tutta
- I computer sono macchine per il trattamento dell'informazione e l'informazione è costituita per la maggior parte da testi, che a loro volta sono composti da caratteri

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
87



Campo di variabilità

- Intervallo finito

Tipo	Dimensione (byte)
<b>Char</b>	<b>1</b>

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
88

## Codifiche binarie

- Ogni carattere è rappresentato da uno specifico codice binario:
  - ad ogni carattere corrisponde una rappresentazione numerica univoca
- Le codifiche binarie più diffuse nel mondo informatico sono:
  - Codifica ASCII
    - (American Standard Code for Information Interchange)
  - Codifica EBCDIC
    - (Extended Binary Coded Decimal Interchange Code)

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    89

## Codice ASCII ...

- Rappresenta 128 simboli diversi (codice di 7 bit ): lettere dell'alfabeto, cifre, segni di punteggiatura e altri simboli

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    90

## Codice ASCII

- 'A' è maggiore di ';', che è maggiore di '&'
- Si ha che:
  - a < b < c < ..... < z
  - A < B < C < ..... < Z
  - 0 < 1 < 2 < ..... < 9


Carattere	Decimale	Binario
{	123	1111011
a	97	1100001
A	65	1000001
B	66	1000010
;	59	0111011
3	51	0110011
&	38	0100110

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    91

## Costanti Carattere

- Una costante carattere è un carattere racchiuso fra apici singoli
- Il valore della costante è il valore che la macchina associa a quel carattere nella particolare codifica usata (solitamente ASCII)
- |     |     |     |      |
|-----|-----|-----|------|
| 'A' | 'x' | '0' | '\$' |
|-----|-----|-----|------|
- Esistono alcune costanti particolari (in Windows):
  - New line (lf)                    '\n'
  - Carriage return (cr)           '\r'
  - Backspace (bs)                '\b'
  - Horizontal tab (tab)           '\t'
  - Form feed (ff)                '\f'
  - Backslash (\)                '\\'

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    92



## Metodi di input/output

```
char x;
```


Output

```
printf("%c", x);
```

Input

```
scanf ("%c", &x);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    93




## ... approfondimento

- In C il tipo char è un tipo intero di 1 byte utilizzato principalmente per rappresentare caratteri

```
#include<stdio.h>
main( ) {
    char x=65;
    printf("%c", x); // stampa A
    printf("%d", x); // stampa 65
    x='A';
    x=x+1;
    printf("%c", x); // stampa B
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    94




## Esercizio

- Scrivere un programma che legge prima due caratteri e poi li stampa per due volte in ordine inverso.

• Esempio:

```
• ab      →      baba
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    95




## Soluzione

```
#include<stdio.h>
main( ) {
    char x, y;
    printf("Digita due caratteri: ");
    scanf("%c%c", &x, &y);
    printf("Hai digitato per 2 volte (ordine inverso):");
    printf("%c%c%c%c", y, x, y, x);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    96






Esercizio

- Scrivere un programma che legge prima un carattere e poi stampa i due caratteri che lo precedono
- Esempio:
  - Se leggo d stampa: cb

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
97




Soluzione

```

#include<stdio.h>
main( ) {
    char x;
    printf("Digita un carattere: ");
    scanf("%c", &x);
    // x=x-1;
    // printf("%c", x);
    printf("%c", --x);
    // x=x-1; printf("%c", x);
    printf("%c", --x);
}
  
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
98



Esercizio

- Scrivere un programma che legge prima tre caratteri e poi li stampa in ordine inverso
- Esempio:
  - Se leggo CDE stampa: EDC

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
99




Soluzione

```

#include<stdio.h>
main( ) {
    char x, y, z;
    printf("Digita tre caratteri: ");
    scanf("%c%c%c", &x, &y, &z);
    printf("Hai digitato (ordine inverso): ");
    printf("%c%c%c", z, y, x);
}
  
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
100



## Esercizio

- Scrivere un programma che legge prima un carattere e poi stampa il carattere che lo segue e quello che lo precede
- Esempio:
  - Se leggo D stampa: E C


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
101



## Soluzione

```
#include<stdio.h>
main() {
    char x;
    printf("Digita un carattere: ");
    scanf("%c", &x);
    x=x+1;
    printf("Il succ = %c ", x);
    x=x-2;
    printf("Il prec = %c\n", x);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
102



## Esercizio

- Scrivere un programma che, dato un carattere, costruisce un triangolo di altezza 3 con tale carattere
- Input:
 


v
- Output:
 

v

vv

vvv

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
103




## Operatori di incremento e decremento

- Operatori unari
- ++ aggiunge uno
- -- sottrae uno

x++
equivale a
x=x+1

x--
equivale a
x=x-1


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
104



## Operatori di incremento e decremento

- Operatori postfissi
- `x++`            prima usa x, poi incrementala
- Operatori prefissi
- `++x`            prima incrementa x, poi usala

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    105




## Operatori di incremento e decremento

- La variabile viene comunque incrementata
- Attenzione ai casi in cui compaiono in istruzioni meno semplici

```
x++;    //equivale a x=x+1;
++x;    //equivale a x=x+1;
y=x++; //equivale a y=x; x=x+1;
y=++x; //equivale a x=x+1; y=x;
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    106



## Operatori di incremento e decremento

- `int n, m=0; n=m++;`  
  • `n=0; m=1`
- `int n, m=0; n=++m;`  
  • `n=1; m=1`

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    107




## Operatori di assegnamento composti

- Il C dispone di operatori di assegnamento che sono combinazioni opportune dell'operatore di assegnamento con operatori aritmetici
  - Esempio:
 

```
a = a + 10;
a += 10;
```
- L'elenco degli operatori di assegnamento è quindi:
 

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


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    108



## Operatore modulo

- Se si vuole conoscere il resto di una divisione tra variabili di tipo intero, è necessario usare l'operatore "modulo" %
- Esempio:
 
$$\begin{aligned} 5 \% 3 &= 2 \\ 3 \% 5 &= 3 \\ 3 \% 3 &= 0 \\ 9 \% 3 &= 0 \end{aligned}$$


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    109



## Conversione di tipo

- Il C consente di comporre in espressioni variabili/costanti di tipi eterogenei
- È possibile assegnare un'espressione che restituisce un tipo ad una variabile di tipo diverso
  - prevede una serie di regole automatiche per la promozione e conversione di tipi in espressioni che contengono variabili/costanti di tipo differente
- Il C consente anche al programmatore di specificare il tipo di dato che vuole ottenere dalla valutazione di espressioni
  - l'operazione di casting per forzare un tipo di dato ad un altro ha la seguente notazione: (tipo) espressione;

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    110



## Casting

- Esempio:
 

```
float f;
int x;
f = 3.14159;
x = (int) f; // si può anche scrivere x = f
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    111



## Conversioni e promozioni implicite

- Prima regola: se un operatore ha operandi di tipi diversi, il tipo "inferiore" è promosso al tipo "superiore"
  - int b=10; float a=2.0;
    - l'espressione a+b restituirà un float
- Seconda regola: nelle espressioni di assegnamento, il risultato dell'espressione (lato destro) è convertito al tipo della variabile (lato sinistro)
  - posso assegnare una variabile/espressione di tipo "inferiore" ad una variabile di tipo "superiore"
- int c=2; float a=c;

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    112

## Esempio

```
#include<stdio.h>
main() {
    int p=257;
    char c;
    printf("%d\n", p);
    c=p;  p=c;
    printf("%d\n", p);
}
```

Stampa:  
257  
1

## Attenzione ...


- Posso anche assegnare, se so cosa sto facendo, una variabile/espressione di tipo superiore ad una variabile di tipo inferiore
  - il compilatore mi segnalerà semplicemente un "warning"
- float c=234.23; int a=c;
  - in questo caso semplicemente tronco... (a vale 234)
- int a=542; char b=a;
  - in questo caso il valore di a verrà erroneamente troncato (char può contenere valori fino a 255)

## Regole di conversione

- Si applicano se nessuno degli operandi è un unsigned
  - se un operando è long double, l'altro è convertito in long double
  - se un operando è double, l'altro è convertito in double
  - se un operando è float, l'altro è convertito in float
  - char e short sono convertiti in int
  - se uno degli operandi è long (int), l'altro è convertito in long
- Un'eccezione (recente) alle regole precedenti:
  - in un'espressione che coinvolge float e double, i float non sono promossi a double

## Operatori relazionali


- Gli operatori relazionali sono:
  - <    <=    >    >=
- a cui si associano gli operatori di uguaglianza:
  - ==    !=
- Da notare che gli operatori relazionali hanno una precedenza inferiore a quelli aritmetici, per cui nella relazione  $x < y-1$  viene valutata dapprima la relazione  $y-1$  ed il risultato viene confrontato con  $x$ , ovvero la relazione assume la forma:
  - $x < (y - 1)$



## Operatori logici

- Gli operatori logici sono costituiti da due operatori binari: && (and) e || (or) e da un operatore unario ! (not)
- Le espressioni con operatori logici vengono sempre valutate da sinistra a destra ed il risultato è pari ad 1 se la relazione è vera, 0 altrimenti
- L'operatore unario ! converte un operando non nullo in uno pari a 0, ed un operando nullo in uno pari a 1


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    117



## Perché usare costanti?

- Per evitare di scrivere più volte in un programma un'espressione che rappresenta un numero, per esempio, quando è molto complicata, o per garantire che non ci siano difformità tra le varie occorrenze
- Per migliorare la leggibilità dei programmi: per esempio, usare sempre una costante per pigreco
- Per riutilizzare i programmi
  - per esempio, un programma che utilizza il valore 100 (ad esempio manipola matrici quadrate di dimensione 100) può essere facilmente riutilizzato quando il valore da utilizzare è uguale a 200 (ad esempio per le matrici di dimensione 200), se tale valore è rappresentato da una costante


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    118



## #define

- Costanti definite mediante direttiva al preprocessore  
`#define Nome costante`
- Semantica:
  - Tutte le occorrenze di Nome (purché non siano racchiuse tra apici e non facciano parte di un'altra stringa) vengono rimpiazzate con costante
  - Nomi delle costanti scritti con caratteri maiuscoli (per convenzione)
  - Dopo il `#define` non serve il ;

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    119



## Esempio


```
#define SIZE 10

int i=SIZE;
```

- Viene tradotto dal preprocessore in

```
int i=10;
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    120



Costanti

```

/* Calcolo area cerchio */
#include <stdio.h>
#define PI_GRECO 3.14


main() {
    float raggio, area;

    printf("Digita il raggio: ");
    scanf("%f", &raggio);
    area = raggio*raggio*PI_GRECO;
    printf("Area: %f", area);
}

```

La definizione di costante  
implica che il suo valore non può  
essere modificato

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
121



Esempio

```


#include<stdio.h>
#define SIZE 3

main() {
    int x=SIZE;
    int y= SIZE+2;
    printf("%d %d", x, y);
}

```

Stampa:  
**3 5**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
122



Esempio

```

#include<stdio.h>
#define SIZE 3

main() {
    printf("SIZE");
}


```

**Semantica**

Tutte le occorrenze (**purché non siano racchiuse tra apici** e non facciano parte di un'altra stringa) vengono rimpiazzate con **costante**

Stampa: **SIZE**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
123



Esempio

```

#include<stdio.h>
#define SIZE 3

main() {
    int MYSIZE = 2;
    int y = MYSIZE;
    printf("%d", y);
}

```

**Semantica**

Tutte le occorrenze (purché non siano racchiuse tra apici e **non facciano parte di un'altra stringa**) vengono rimpiazzate con **costante**

Stampa: **2**

**SIZE** parte di **MYSIZE**, quindi non si rimpiazza con **3**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
124



### Esercizio

- Scrivere un programma che riceve in input un intero  $n$  di quattro cifre e stampa la somma delle cifre di  $n$ . Ad esempio, se  $n = 1205$  allora il programma stampa 8
- Usare la dichiarazione di costante



### Esercizio

- Data una sequenza di 4 valori compresi tra 0 e base-1, calcolare il corrispondente valore decimale
- Se viene immesso in input: 0101

output  $1 \cdot (2^0) + 1 \cdot (2^2)$ , pari a 5, se la base è 2

output  $1 \cdot (10^0) + 1 \cdot (10^2)$ , pari a 101, se la base è 10

output  $1 \cdot (8^0) + 1 \cdot (8^2)$ , pari a 65, se la base è 8