

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

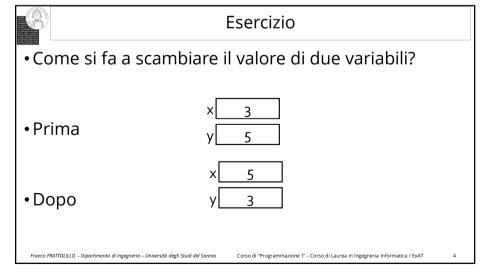
Esercizio

• 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);
}

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

Eso	ercizio
Determinare i valori di p, q seguenti istruzioni:	ed r al termine delle
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 Cor	so di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT 3



```
Esercizio: scambio variabili

• È corretto questo programma?

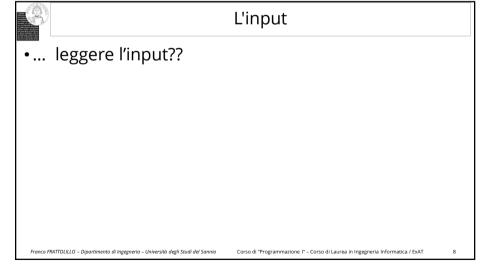
#include <stdio.h>
main() {
    int x, y;
    x=3; y=5;
    x=y; y=x;
    printf("%d\n", x);
    printf("%d\n", y);
}

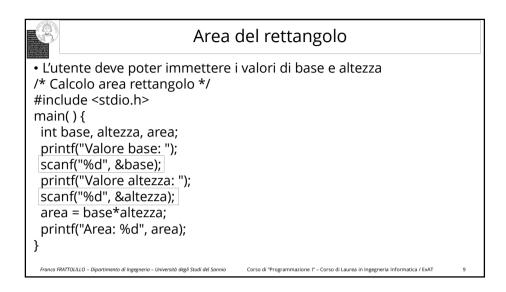
Franco FRATTOLLLO - Dipartimento di ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / Exat 5
```

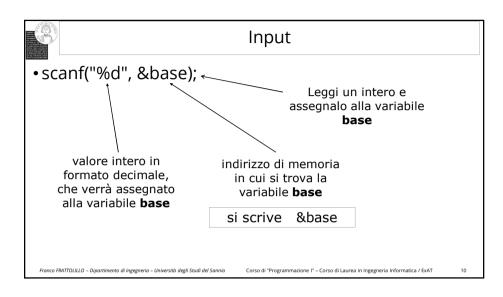
```
#include <stdio.h>

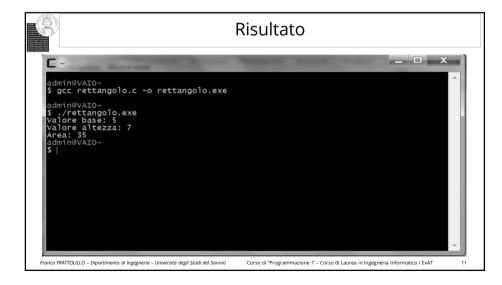
main() {
    int x, y, temp;
    x=3; y=5;
    temp=x;
    x=y;
    y=temp;
    printf("%d\n", x);
    printf("%d\n", y);
}

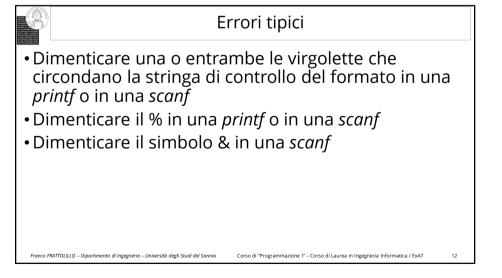
france FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 6
```











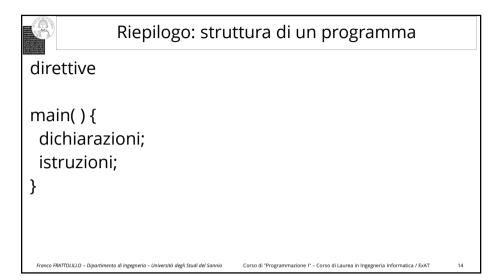


... abbiamo visto

- 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





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

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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 Sanni



... 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;

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sanr

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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" ?

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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



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



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

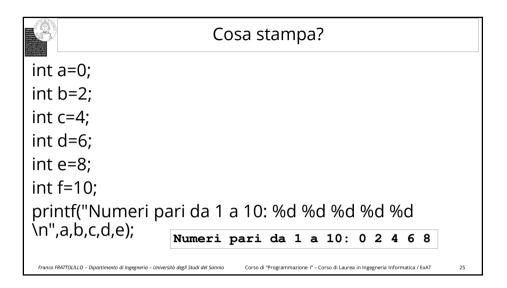
Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

23

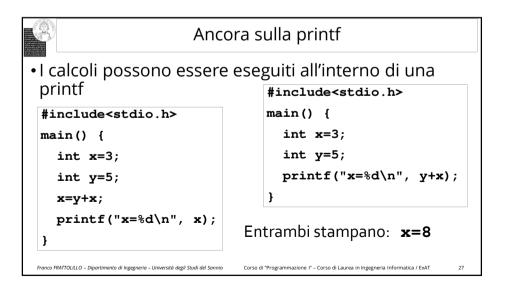
Esempi di printf

```
#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
```



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

Franco FRATTOLILLO - Diportimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione l" - Corso di Laurea in Ingegneria Informatica / Exat 26
```



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 - Diportimento di Ingegneria - Università degli Studi del Sonnio

Corso di "Programmazione l' - 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



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



Esercizio

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

Prove

tecniche di

visualizzazione

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sanni

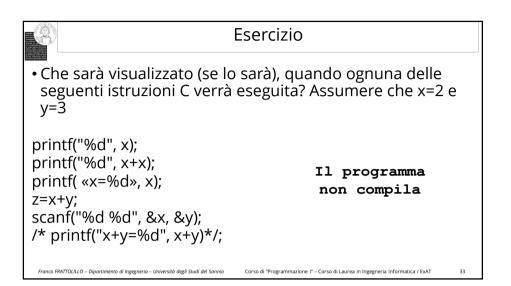
Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

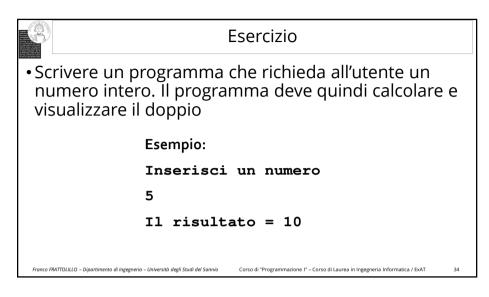


Soluzione

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

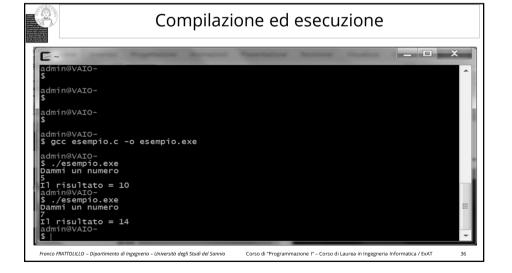
Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sann.





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

Franco FRATTOLILO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione l' - Corso di Laurea in Ingegneria Informatica / EAX 35
```





• 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

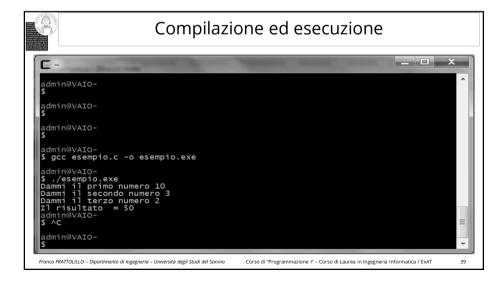
Inserisci il terzo numero 3

Il risultato = 50

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

```
#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", &secondo);
risultato = (primo-secondo-terzo)*primo;
printf("Il risultato = %d", risultato);
}

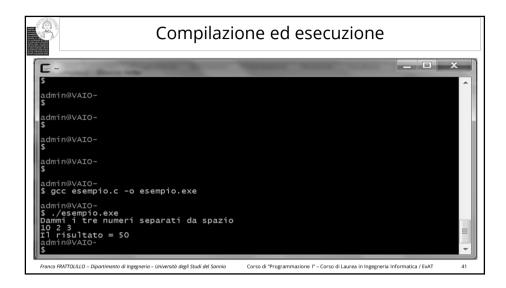
**Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione 1" - Corso di Laurea in Ingegneria Informatica / EAT 38
```



```
#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);
}

##Include<stdio.h>
##Include<stdio.h

##Include<s
```



```
• 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);
}

*Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione l" - Corso di Laurea in Ingegneria Informatica / Exat 42
```

```
#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);
}

Franco FRATTOLLLO - Dipartimento di Ingegneria - Liniversità degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 43
```

```
• 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");
}
```

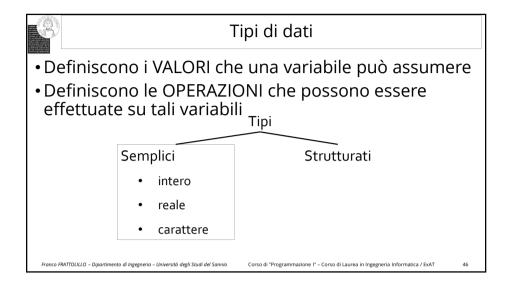


Soluzione

- 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





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

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sanni

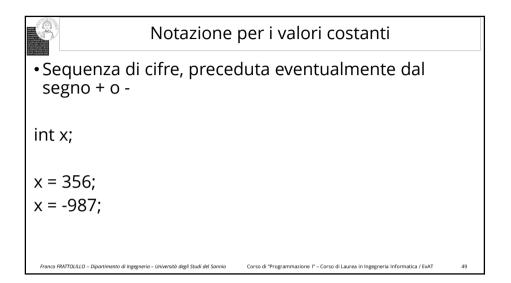
Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

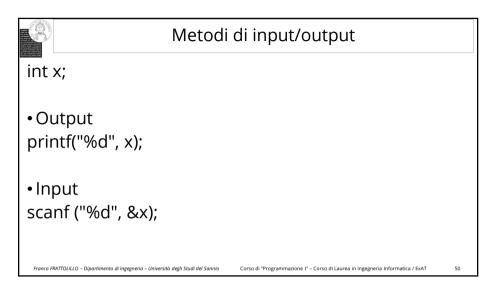
Campo di variabilità

Intervallo finito

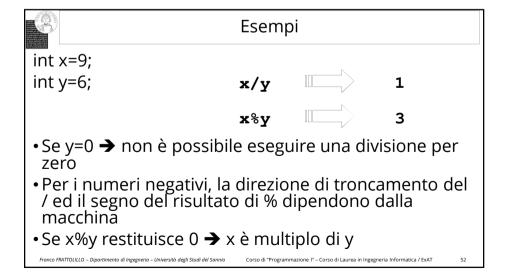
Tipo	Dimensione (byte)	Valore minimo	Valore massimo
short int	2	-32768	+32767
int	4	-2 ³¹	2 ³¹ - 1
long int	8	-2 ⁶³	2 ⁶³ - 1

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio





	Operazioni	
of the interaction wheel also the face with the files		
+	somma	
-	meno unario	
-	differenza	
*	prodotto	
/	divisione intera	
%	resto della divisione intera (modulo)	
Seemen FR	ATTOURG Discriment of transports. Helpsychia dell'Establed Foreits. Corrold Theory managing II. Corrold Lawre in Inspension Information (FuAT	F1
Franco FR	ATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio Corso di "Programmazione l" – Corso di Laurea in Ingegneria Informatica / ExAT	51





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
unsigned short int	2	0	65535
unsigned int	4	0	+2 ³² - 1
unsigned long int	8	0	+2 ⁶⁴ - 1

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



short e long

- Sono <u>qualificatori</u> 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 */

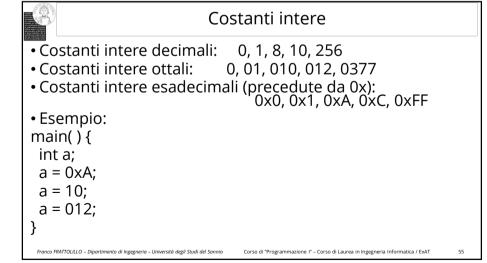
• short a; /* int può essere omesso */

• long utilizza un maggior numero di bit per rappresentare gli interi

• long int a; /* intero a 64 bit */

• long a; /* int può essere omesso */

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio



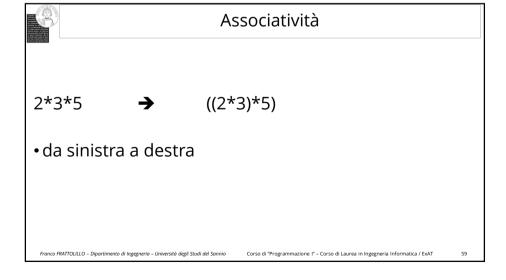


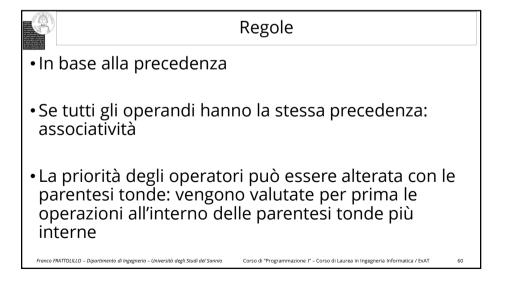
Precedenze

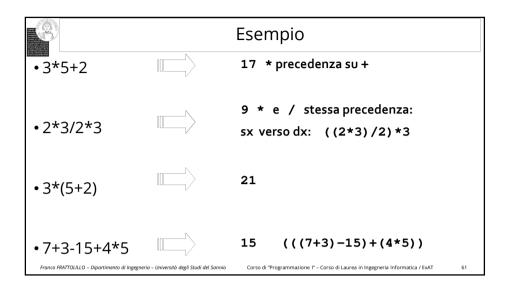
- Il C dispone di un insieme di regole che determinano l'ordine in cui le varie operazioni devono essere eseguite
- Per le operazioni aritmetiche le precedenze sono quelle definite in matematica

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

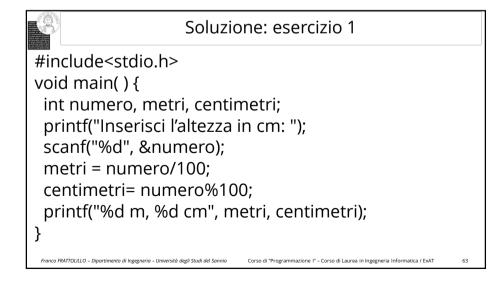
	denza degli operatoi	
Operatori	Associatività]
Parentesi:	dall'interno all'esterno	+ alta
Operatore unario:	da destra a sinistra	
Operatori binari: */%	da sinistra a destra	
Operatori binari:	da sinistra a destra	+ bassa

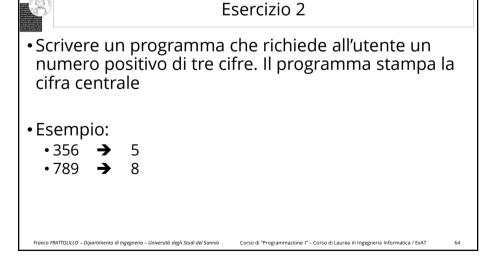






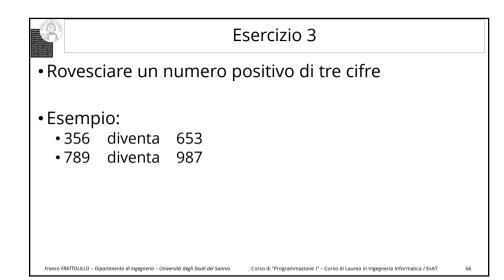






```
#include<stdio.h>
main() {
  int n, cfr;
  printf("Inserisci un numero positivo di tre cifre: ");
  scanf("%d", &n);
  cfr = (n/10) % 10;
  printf("Cifra centrale: %d", cfr);
}

Franco FRATTOLLLO - Dipartimento di Ingegeneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegeneria Informatica / ENAT 65
```



```
#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);
}

Franco FRATTOLLLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione l" - Corso di Laurea in Ingegneria Informatica / ENAT 67
```

```
#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);
}

**Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / EAAT 68
```



- 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

```
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
```



Tipo reale

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



Campo di variabilità

Tipo	Dimensione (byte)	Valore minimo	Valore massimo
float precisione singola	4	-3.2·10 ^{±38}	+3.2·10 ^{±38}
double precisione doppia	8	-1.7·10 ^{±308}	1.7·10 ^{±308}



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 -3·10³ cioè -3000 5e-2 rappresenta 5·10⁻² cioè 0.05

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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



Funzioni aritmetiche

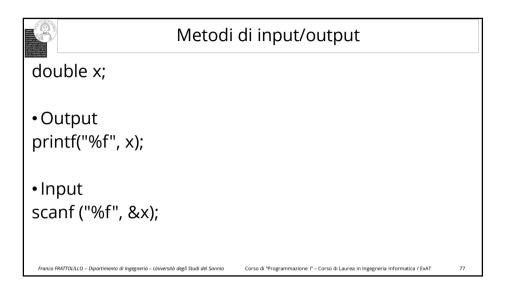
- #include<math.h>
- x e y di tipo double e restituiscono un double
- pow(x, y)
- sin(x) seno di x, con x espresso in radianti
- cos(x) coseno di x, con x espresso in radianti
- exp(x)
- log(x) logaritmo naturale di x radice quadrata x, x>=0 sqrt(x) logaritmo in base 10 di x
- log10(x)

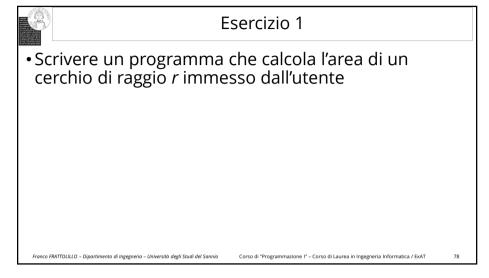


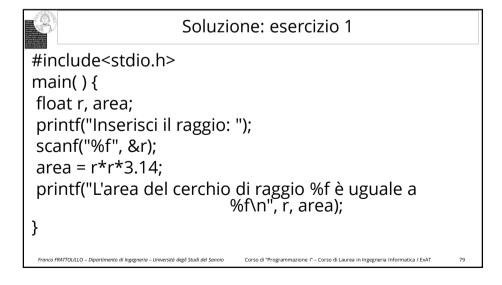
Metodi di input/output

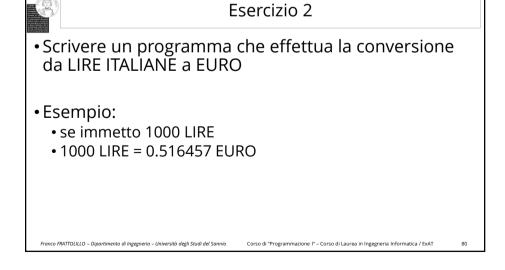
float x;

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









```
#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);
}
```

```
#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);
}

Franco FRATTOLLLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione l' - Corso di Laurea in Ingegneria Informatica / ExAT 82
```

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

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sanni

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

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);
}

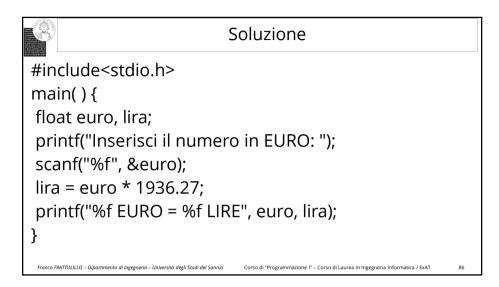
Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / EAT 84
```



- 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





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

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sann

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



Campo di variabilità

Intervallo finito

Tipo	Dimensione (byte)
Char	1

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio



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

orso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / I

AT 8



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



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
В	66	1000010
;	59	0111011
3	51	0110011
&	38	0100110

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

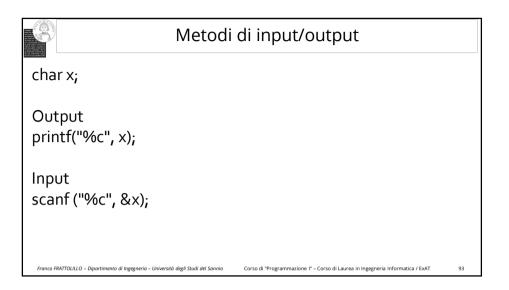
Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

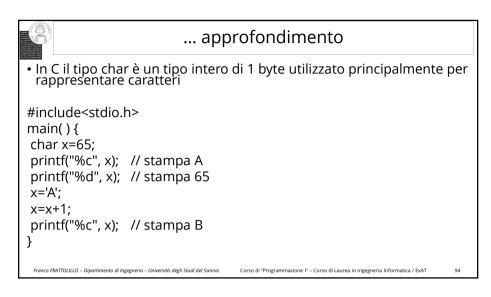


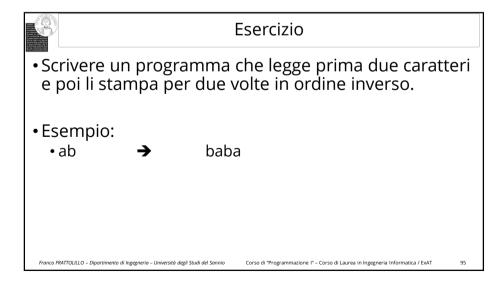
Costanti Carattere

- 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 Sant







```
#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%c", y, x, y, x);
}

**Fanco FRATTOLILLO - Dipartimento di Ingegnerio - Università degli Studi del Sannio

**Corso di "Programmazione 1" - Corso di Laurea in Ingegneria Informatica / Exat**

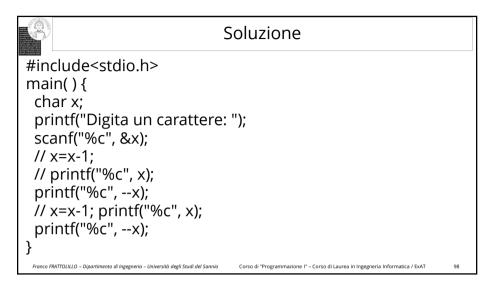
96
```



- 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





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 San

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

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);
}

#include<stdio.h>

#include<stdio.h

#include<stdio.h>

#include<stdio.h

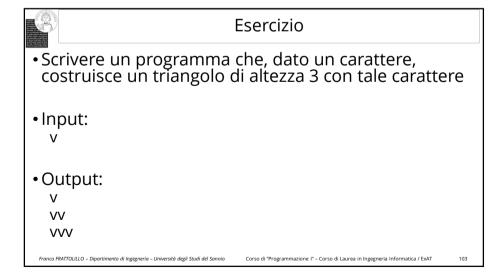
#include<st
```



- 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

```
#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);
}
```



```
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



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



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 Sanni

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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



Operatore modulo

- Se si vuole conoscere il resto di una divisione tra variabili di tipo intero, è necessario usare l'operatore "modulo" %
- Esempio:

5%3 = 2

3%5 = 3

3 % 3 = 0

9 % 3 = 0

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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 <u>promozione</u> e <u>conversione</u> 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 <u>casting</u> 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



Casting

• Esempio:

float f; int x:

f = 3.14159;

x = (int) f; // si può anche scrivere <math>x = f

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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 Sann

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

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

Stampa:
  257
  1

Franco FRATTOLLLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ENAT

113
```



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)

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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)

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



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 / ExA

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

ranco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sannio

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

110



#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 Sann

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT

Esempio

#define SIZE 10

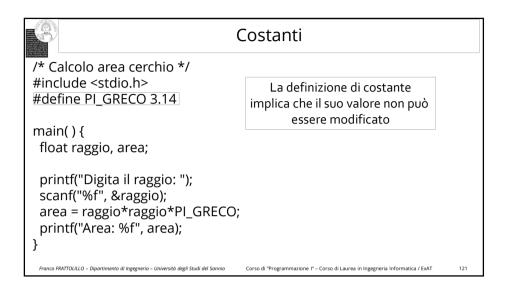
int i=SIZE;

• Viene tradotto dal preprocessore in

int i=10;

Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli Studi del Sann

Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT



```
#include<stdio.h>
#define SIZE 3

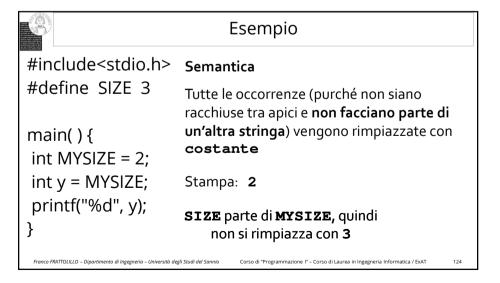
main() {
  int x=SIZE;
  int y= SIZE+2;
  printf("%d %d", x, y);
}

France FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Esempio

Stampa:
  Stampa:
  3 5
```

	Esempio
#include <stdio.h></stdio.h>	Semantica
#define SIZE 3	Tutte le occorrenze (purché non siano racchiuse tra apici e non facciano parte
main() { printf("SIZE"); }	di un'altra stringa) vengono rimpiazzate con costante
	Stampa: SIZE
Franco FRATTOLILLO – Dipartimento di Ingegneria – Università degli St.	udi del Sannio Corso di "Programmazione I" – Corso di Laurea in Ingegneria Informatica / ExAT 123



126



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

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT



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 $1*(2^0) + 1*(2^2)$, pari a 5, se la base è 2 output $1*(10^0) + 1*(10^2)$, pari a 101, se la base è 10 output $1*(8^0) + 1*(8^2)$, pari a 65, se la base è 8

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio