

Lezione: questa Settimana

MER 8-10

GIOV 16-18 \hookrightarrow Scrivere con geometria

VEN 10-11

Settimana 20-24 Ottobre

MER 22/10 8-10 geometria

VEN 24/10 10-11 NO LAB CALC

C: tipi di dati \Rightarrow dichiarazione variabili:

linguaggio struttura

funzioni:

struttura dati

funzioni

$a = \sqrt{3}$

dati
input

3: input

dati
output

a: valore di output

file sorgente nome.c

file di dati scritto con editor
di testo

```
int main() {  
    printf("Primo programma\n");  
}
```

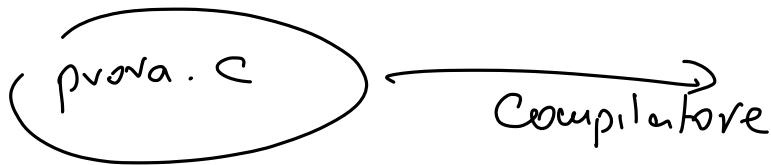
prova.c

printf è funzione di stampa sullo schermo
f sta per formattato

"primo programma\n" è stringa di caratteri

' Ciao' } " Ciao" } stringhe di caratteri:

In : Vai a capo



Codice binario
non eseguibile



Codice binario
eseguibile

Compilazione & gcc

gcc -c prova.c => crea prova.o
non eseguibile

gcc -o prova.exe prova.c => crea prova.exe
eseguibile

gcc -o pippo prova.c => pippo eseguibile

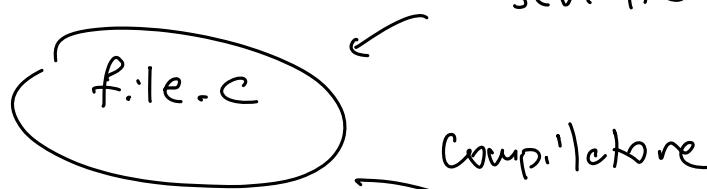
./prova.exe & esegue il programma

primo programma D → prompt scrive in nel testo

/ & slash

\ & back-slash

scrivo da voi:



library mettere in



eseguibile

uso funz. matematiche

prova.c

```
1 #include <math.h>
2
3 int main() {
4     double x;
5     x = sqrt(2.4);
6 }
7 }
```

Istruzione del precompilatore

math.h: Header file delle lib. matematica

contiene dichiar. di type e funz. e costanti matem.

sqrt: calcolo radice quadrata

gcc -o prova.exe

prova.c

-lm

uso librerie mat.

nel linker

per fare esegibile

Compil.

→ prova.o

Linker

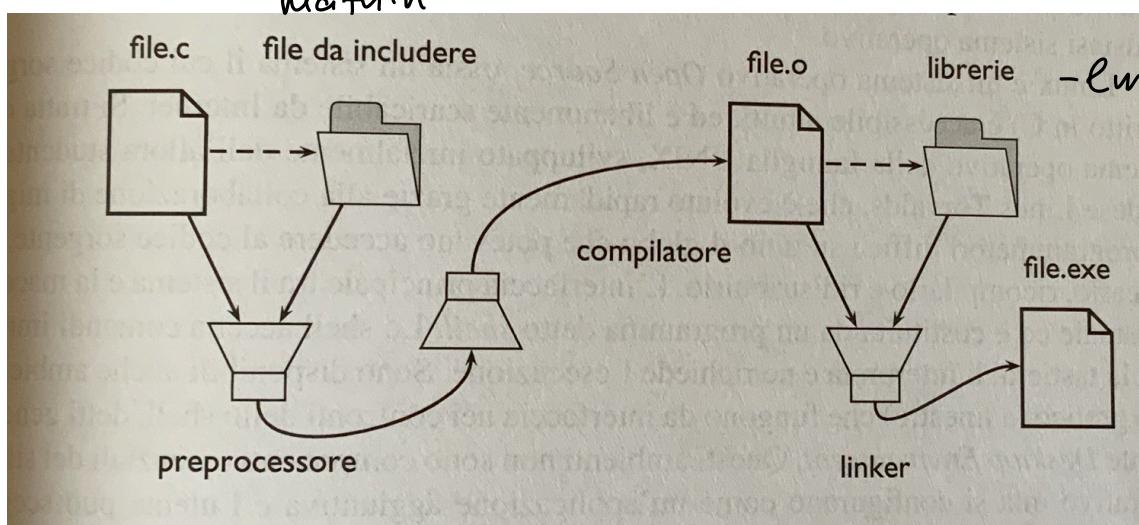
→ prova.exe

precompilazione

prova.c

temp.c
migliorie
di rispe
perche'
include
math.h

stdio.h
math.h



Comando	Commento
gcc -Wall -c prova.c	compila il file sorgente prova.c e crea il file oggetto prova.o Non c'è nessun eseguibile
gcc -Wall -o app.exe prova.c	compila e linka il file sorgente prova.c e crea il file eseguibile app.exe Non c'è nessun file.o in questo caso
gcc -Wall -lm -o app.exe prova.c	compila e linka il file sorgente prova.c utilizzando la libreria matematica (opzione -lm) e crea il file eseguibile app.exe Non c'è nessun file.o in questo caso
gcc -Wall -o app.exe prova.c grafica.o	compila il file sorgente prova.c e crea il file eseguibile app.exe Utilizzando anche il contenuto del file grafica.o (compilato in precedenza)
gcc -Wall -o app.exe prova.c grafica.c	compila i due file sorgente prova.c e grafica.c e crea il file eseguibile app.exe Non viene creato nessun file oggetto

-Wall : w: warning all: tutte

Sintassi: di base

prova.c

int main() {

 int n; *Dichiev. n*

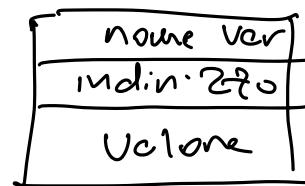
 double a,b;

Dich. a,b

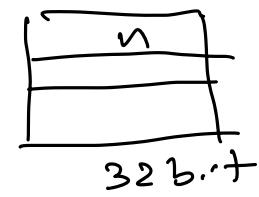
}

gcc -o app.exe prova.c
./app.exe

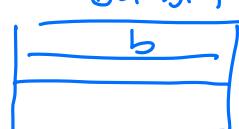
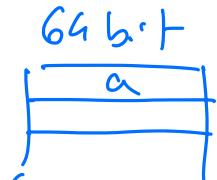
ogni istruzione deve
finire con ;



[int main() { int n; double a,b; }



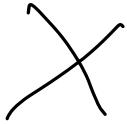
int main() {
int n;
double a,b;
}



int main() {

a, b;

}

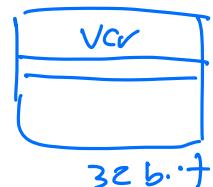
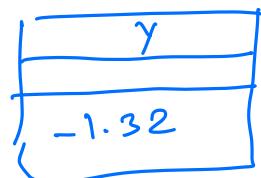


Non Corretta

Tipo	Variabile	Bit
char	carattere	8
short	intero	16
int	intero	32
long	intero	32
float	razionale	32
long long int	intero	64
double	razionale	64
long double	razionale	96

Dichiaraz. float var; ricorda che a 32 bit

Dichi: float y = -1.32;
+ assegnazione



Nome Variabili:

stringhe alfanumeriche

senza uso di caratteri

speciali:

Nome valido	Nome non valido
Vel	Velocita'
_1F	1F
xyzzy	xyz:zy
x4756	.4756
--a	+a

non utilizzabili: . , ; : + - = / * < > ! ? ~ @ # \$ % ^ & | " ` !

- underscore diviso -

name - cognome ✓

name - cognome ~~✓~~

Operazione	Simbolo	Esempio
somma	+	$a + b$
differenza	-	$a - b$
prodotto	*	$a * b$
quoziente	/	a / b

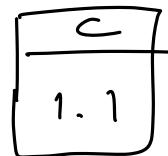
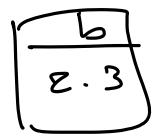
double $a, b;$

$a = -1.2;$

$b = 2.3;$

double $c = a + b;$

$c = a + b;$



Operatori

uniari
binari:

- 3

$a - b$

LHS
Left
Hand
Side

- o operatore unario

- o operazione binaria

RHS

Right
Hand
Side

3!

Operatore Relazione

>	maggiore di
>=	maggiore o uguale a
<	minore di
<=	minore o uguale a
==	uguale a
!=	non uguale a

$a = b;$

assegnazione

$a == b;$

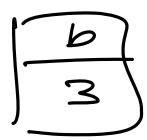
uguaglianza

$a = b:$ valore di a diretta uguale a b .

$a = 2;$



$b = 3;$



$a = b;$

valore di b assegnato a a

$b = a;$

valore di a viene assegnato a b



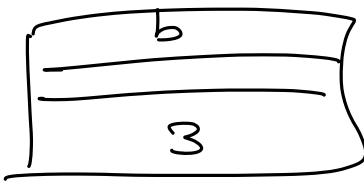
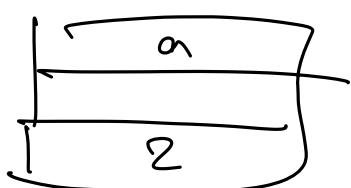
Ugualmente \neq

$a == b$

vero se a, b hanno lo stesso
valore

falso se a, b diversi

$a = 2; b = 3;$



$a == b$ FALSO

$b == a$ FALSO

Variabili booleane sono es: bool.

JSone int, Char;

int decisione = $a == b$;

LHS

RHS



32 bit

decisione = $a != b$;



decisione = $a > b$;



- chiedere angolo all'utente
- prendere rettangoli in gradi.
- convertire in rad;
- stampare sullo schermo

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

```
int main() {
```

printf("inserisci angolo in gradi: ");

double angolo;

scanf("%lf", &angolo);



↳ e-commerce

descrizione verificabile lf: long float

doub)le radice:

radice = (angolo/180.) * M_PI;

printf("gradi: %lf, radice: %lf\n", angolo, radice);

Convert. C

gcc -o app.exe convert.c -lm

app.exe

inserisci angolo in gradi: 180

library

Standard input output

gradis: 180, radis: 3.14----

float a = 1.2;

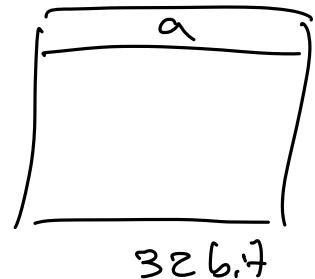
~~printf("%f", a);~~

printf("%f", a); 1.2

printf("a=%f", a); a = 1.2

printf("a=%d", a);

descritt. inter:



Input con printf()

input con scanf()

Descrittore	Tipo
%f, %e, %g	float, double
%d, %i	int
%c	char (singolo)
%s	char (stringa)

Descrittore	Tipo
%f	float
%lf	double
%Lf, %lld	long double
%d, %i	int
%u	unsigned int
%Lu	unsigned long long int
%c	char (singolo)
%s	char (stringa)

int a, b, c;

a = 8;

b = 4;

c = a/b; \emptyset

double d = a/b; d = ? \emptyset
div. tra inter:

double d = 3/4; \neq

d = 3.14; 0.75

rationale / intero

automatamente retz / rcz

double d = 3.14
 3/4. } 0.75
 3.14. }

3/4 0

int a=2, b=3;

double d = a/b; \neq

d = (double)a/b; 0.666666c-

rcz / intero

\Rightarrow retz / rcz

(double) a : casting di a come double

a / (double) b 0.666

(double) a / (double) b 0.666

(double)(a/b) \neq