



Insegnamento di Elementi di informatica (6 CFU)

Il linguaggio C - Parte 2 bis

Nadia Ranaldo
RCOST - Dipartimento di Ingegneria
Università degli Studi del Sannio

Perché usare costanti?



- per non 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 differenze tra le varie occorrenze*
- permette di capire meglio i programmi, *per esempio usare sempre una costante per pigreco*

Costanti



```
/* Calcolo area cerchio */
```

```
#include <stdio.h>
#define PI_GRECO 3.14

main()
{
    float raggio, area;

    printf("Dammi 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

NOTA

Le direttive non terminano con il ;

Costanti in C



1. Costanti testuali (#define)
2. Variabili read only (const)

#define



- **Sintassi**

#define NomeVariabile costante

- **Semantica**

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

5

Esempio



```
#define SIZE 10  
int i=SIZE;
```

Viene tradotto dal preprocessore in

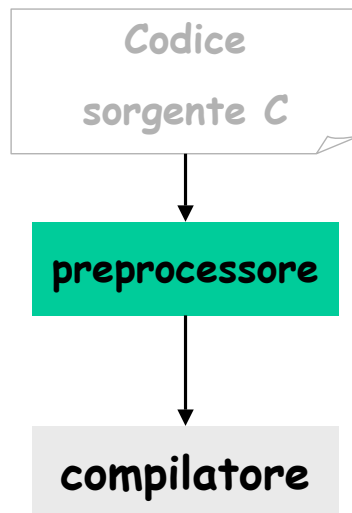
```
int i=10;
```

- **Note:**

- Nomi delle costanti scritti con caratteri maiuscoli (per convenzione)
- Dopo alla fine **#define** non serve il ;

6

Preprocessore



7

Cosa fa il preprocessore?



1. Processa le cosiddette **direttive al preprocessore:**

- inclusione di file

```
#include <nomefile>
```

- prende il file **nomefile** e lo inserisce al posto della direttiva

- definizioni di costanti

```
#define NomeVariabile costante
```

- ogni volta che incontra **NomeVariabile** lo sostituisce con **costante**

2. Elimina i commenti contenuti nel sorgente

8



1. Costanti testuali (#define)

2. Variabili read only (const)



```
#include <stdio.h>
main()
{
    const int A=100;
    printf("%d", A);    //stampa 100
    A=A+1;              // errore
}
```