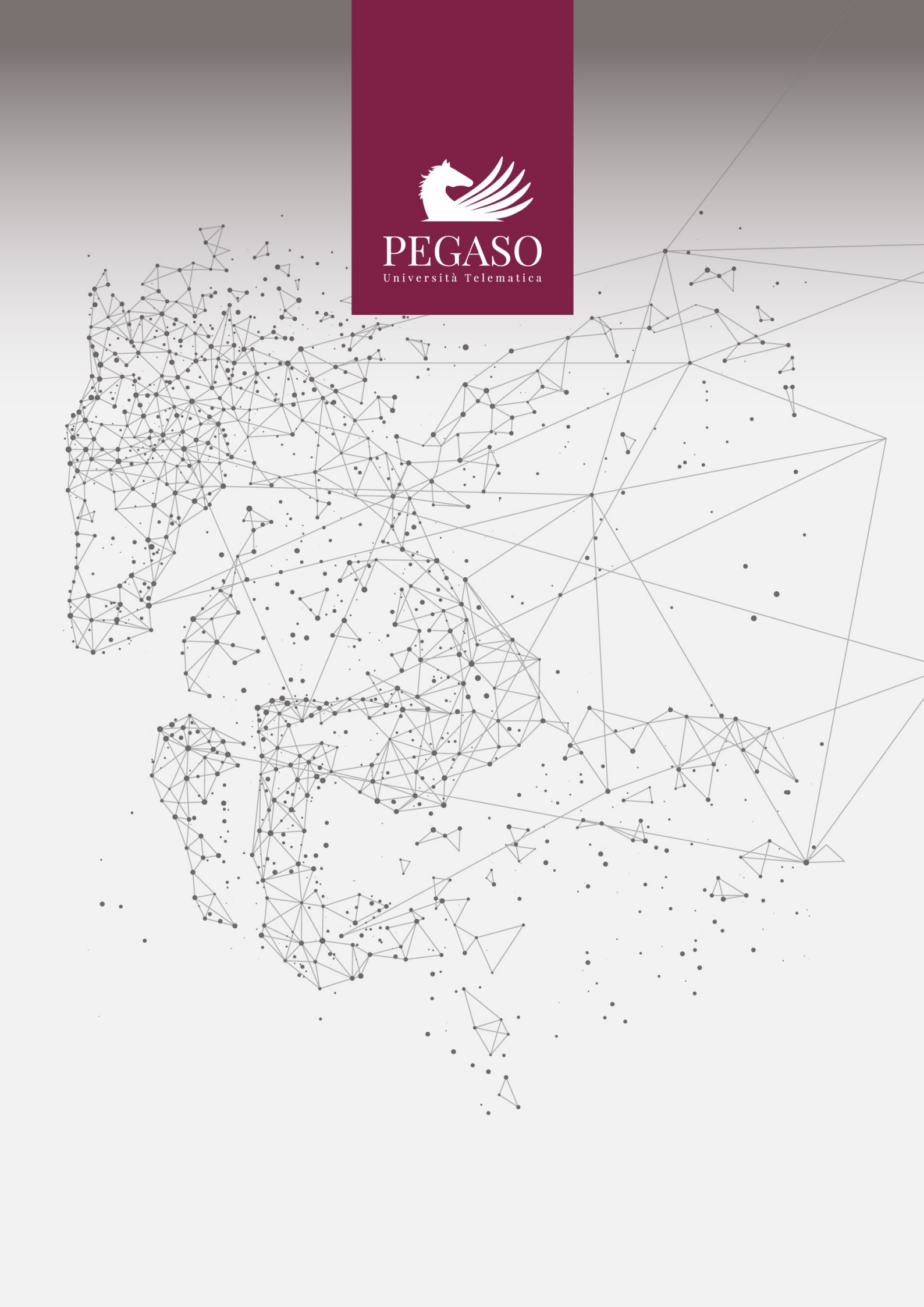
PEGASO
Università Telematica

Indice

1. Definizioni di funzioni

Ogni programma che abbiamo presentato consiste in una funzione chiamata `main` che chiama le funzioni della Libreria Standard per eseguire i suoi compiti.

Vediamo adesso come scrivere nuove funzioni definite dal programmatore.

Funzione square

Considerate un programma che usa una funzione square per calcolare e stampare i quadrati dei numeri interi da 1 a 10.

```
01  // Programma square
02  // Creazione e uso di una funzione definita dal programmatore.
03  #include <stdio.h>
04
05  int square(int y); // prototipo di funzione
06
07  int main(void)
08  {
09     // ripeti 10 volte e ogni volta calcola il quadrato di x
10     for (int x = 1; x <= 10; ++x) {
11        printf("%d ", square(x)); // chiamata della funzione
12     }
13
14     puts("");
15  }
16
17  //  definizione della funzione square
18  int square(int y)
19 // nota: y e' una copia dell'argomento della funzione
20  {
```

```
21      return y * y;
22      // restituisce il quadrato di y come un valore int
21  }
```

Output del programma:

```
1 4 9 16 25 36 49 64 81 100
```

Chiamata della funzione square

La funzione `square` è invocata o chiamata nella funzione `main` all'interno dell'istruzione `printf` (riga 11).

```
printf("%d ", square(x)); // chiamata della funzione
```

La funzione `square` riceve una copia del valore dell'argomento `x` nel parametro `y` (riga 18).

Poi `square` calcola `y * y` e restituisce il risultato alla riga 11 nella funzione `main` dove `square` è stata invocata (riga 11).

La riga 11 continua passando il risultato `square` alla funzione `printf`, che stampa il risultato sullo schermo.

Questo processo è ripetuto 10 volte, una volta per ogni iterazione dell'istruzione `for`.

Definizione della funzione square

La definizione della funzione `square` (righe 18–21) evidenzia che `square` si aspetta un parametro intero `y`.

La parola chiave `int` che precede il nome della funzione (riga 18) indica che `square` restituisce un risultato intero.

L'istruzione return in `square` restituisce il valore dell'espressione `y * y` (cioè il risultato del calcolo) alla funzione chiamante.

Prototipo della funzione square

La riga 5

```
int square(int y); // prototipo di funzione
```

è un prototipo di funzione (chiamato anche dichiarazione di funzione).

L'`int` tra parentesi informa il compilatore che `square` si aspetta di ricevere un valore intero dalla funzione chiamante.

L'`int` alla sinistra del nome della funzione `square` informa il compilatore che `square` restituisce alla funzione chiamante un risultato intero.

Il compilatore fa riferimento al prototipo della funzione per controllare che in ogni chiamata alla funzione `square` (riga 11):

- il numero di argomenti sia corretto,
- i tipi degli argomenti siano corretti,
- i tipi degli argomenti siano in ordine corretto,
- il tipo di ritorno sia coerente con il contesto nel quale è chiamata la funzione.

Formato della definizione di una funzione

Il formato della definizione di una funzione è

tipo-del-valore-di-ritorno nome-della-funzione (lista-dei-parametri)

{

istruzioni

}

Il nome della funzione è qualsiasi identificatore valido.

Il tipo del valore di ritorno è il tipo del risultato restituito alla funzione chiamante.

Il tipo di valore di ritorno `void` indica che una funzione non restituisce alcun valore. Insieme, il tipo del valore di ritorno, il nome della funzione e la lista dei parametri costituiscono la cosiddetta intestazione della funzione.

La lista di parametri è un elenco separato da virgole che specifica i parametri che la funzione riceve quando è chiamata.

Se una funzione non riceve alcun valore, la lista di parametri è `void`.

Per ogni parametro deve essere specificato esplicitamente un tipo.

😊 *Prevenzione di errori*

Controllate che le vostre funzioni che devono restituire valori lo facciano.

Controllate che le vostre funzioni che non devono restituire valori non lo facciano.

☹ *Errore comune di programmazione*

Specificare parametri di funzione dello stesso tipo come

```
double x, y
```

invece di

```
double x, double y
```

genera un errore di compilazione.

☹ *Errore comune di programmazione*

Porre un punto e virgola dopo la parentesi destra che chiude la lista dei parametri della definizione di una funzione è un errore di sintassi.

☹ *Errore comune di programmazione*

Ridefinire un parametro come variabile locale in una funzione genera un errore di compilazione.

😊 *Buona pratica di programmazione*

Sebbene non sia scorretto fare così, non usate gli stessi nomi per gli argomenti di una funzione e per i corrispondenti parametri nella definizione della funzione. Ciò contribuisce a evitare ambiguità.

Corpo della funzione

Le istruzioni all'interno delle parentesi graffe formano il corpo della funzione, che è anche detto blocco.

Le variabili possono essere dichiarate in qualunque blocco e i blocchi si possono annidare (mentre le funzioni non si possono annidare).

☹ *Errore comune di programmazione*

Definire una funzione dentro un'altra funzione è un errore di sintassi.

😊 *Buona pratica di programmazione*

Scegliere nomi significativi per le funzioni e nomi significativi per i parametri rende i programmi più leggibili e contribuisce a evitare un uso eccessivo di commenti.

😊 *Osservazione di ingegneria del software*

Funzioni di piccole dimensioni favoriscono la riutilizzabilità del software.

😊 *Osservazione di ingegneria del software*

I programmi devono essere scritti come collezioni di funzioni di piccole dimensioni. Ciò rende i programmi più facili da scrivere, correggere, mantenere e modificare.

😊 *Osservazione di ingegneria del software*

Il prototipo, l'intestazione e le chiamate di una funzione devono tutti concordare nel numero, nel tipo, nell'ordine degli argomenti e dei parametri, nonché nel tipo del valore di ritorno.

Restituzione del controllo da una funzione

Vi sono tre modi per restituire il controllo da una funzione chiamata al punto in cui tale funzione è stata invocata.

Se la funzione non restituisce alcun risultato, il controllo viene semplicemente restituito quando viene raggiunta la parentesi graffa destra che termina la funzione o quando si

esegue l'istruzione

```
return;
```

Se la funzione restituisce un risultato, l'istruzione

```
return espressione;
```

restituisce alla funzione chiamante il valore di espressione.

Il tipo di ritorno della funzione main

Notate che la funzione `main` ha un tipo di ritorno `int`.

Il valore di ritorno di `main` si usa per indicare se il programma ha completato l'esecuzione correttamente.

In versioni precedenti del C avremmo scritto esplicitamente

```
return 0;
```

alla fine di `main` (0 indica che un programma è stato eseguito con successo).

Il C standard stabilisce che `main` restituisce implicitamente 0 se omettete la precedente istruzione (come abbiamo fatto).

Potete far restituire esplicitamente a main valori diversi da zero per indicare che si è verificato un problema durante l'esecuzione del vostro programma.

Riferimenti bibliografici

- Paul Deitel, Harvey Deitel, "Il linguaggio C – Fondamenti e tecniche di programmazione", Libro edito da Pearson Italia. Include anche utili esercizi di autovalutazione.