

# Laboratorio di programmazione

6 dicembre 2005

## Permutazione

Una *permutazione* su  $n$  elementi è una funzione biiettiva dall'insieme  $\{0, 1, \dots, n - 1\}$  in se stesso. Scrivete una funzione:

```
int conta(int v[], int n, int x)
```

che, dato un vettore  $v$  di lunghezza  $n$ , restituisce quante volte  $x$  compare nel vettore.

Scrivete quindi un programma *nello stesso file sorgente di nome contamain.c* che prende in ingresso  $n$  e un vettore di  $n$  elementi, e che, usando la funzione appena scritta, determina se il vettore rappresenta una permutazione oppure no.

### Esempio di funzionamento

Numero di elementi: <b>5</b> Vettore: <b>1 0 2 4 3</b> Il vettore è una permutazione
--

Numero di elementi: <b>7</b> Vettore: <b>1 0 3 2 4 3 5</b> Il vettore non è una permutazione
--

## Varianti di permutazione

Realizzate le seguenti varianti, verificando ogni volta che il programma continui a essere correttamente compilabile ed eseguibile.

1. Spostate la funzione `conta` in un file sorgente diverso di nome `conta.c`, e aggiungete al file `main.c` contenente il `main` l'intestazione richiesta.
2. Create un file di nome `conta.h` che contiene solo l'intestazione, e includetelo nel file contenente il `main` (al posto dell'intestazione).

3. Riprendete il file `contamain.c` e modificate la funzione `conta` in modo che abbia l'intestazione

```
int conta(int n, int x)
```

e assuma che l'array sia dichiarato esternamente.

4. Operate la stessa modifica nella versione con due file separati, e osservate come dovrete necessariamente definire la variabile `extern` in uno dei due file, e dichiararla nell'altro.

5. Ponete la variabile dichiarata `extern` nell'header file.

## Sequenze di Collatz

Considerate la seguente regola: dato un numero intero positivo  $n$ , se  $n$  è pari lo si divide per 2, se è dispari lo si moltiplica per 3 e si aggiunge 1 al risultato. Quando  $n$  è 1 ci si ferma.

Questa semplice regola permette di costruire delle sequenze: la sequenza che si costruisce a partire dal numero  $n$  è detta *sequenza di Collatz di  $n$* . Ad esempio, la sequenza di Collatz di 7 è:

```
7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

È un noto problema aperto stabilire se ogni sequenza di Collatz termina (cioè, se arriva a 1).

Scrivete innanzitutto una funzione che, dato un numero, dia il successivo in una sequenza di Collatz. Quindi, inseritela in un programma che chiede all'utente un numero e mostra la sequenza di Collatz del numero (con tanto di lunghezza).

### Esempio di funzionamento

Numero: 7 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 Lunghezza: 17
--

Numero: 9 9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 Lunghezza: 20
--

## Ore

Scrivete un programma che, data un'ora sotto forma di tre numeri interi (ore, minuti, secondi), stampi la rappresentazione testuale dell'ora indicata.

### Esempio di input

```
12 30 22
```

### Esempio di output

```
Ore dodici, trenta minuti e ventidue secondi.
```

### Suggerimenti

È importante strutturare bene il programma in funzioni. Un punto di partenza è creare una funzione in grado di stampare la parte delle decine, cioè che su input 2 scriva “venti”, su input 3 scriva “trenta” e così via (usate l’istruzione `switch`). Sulla base di questa è poi possibile creare una funzione che stampi in maniera testuale un numero tra 0 e 99. Si noti però che la prima funzione deve avere un secondo parametro che specifica (a seconda del suo valore) se bisogna stampare l’ultima vocale oppure no (altrimenti numeri come “trentuno” non sarebbero stampabili).