

# Fondamenti di Informatica 2011-2012

## Esercitazione 2

### Abstract

In questa seconda esercitazione ci concentreremo su lettura e stampa di dati di tipo diverso e sui costrutti di selezione e iterazione.

## 1 Informazioni generali

1. Alcune esercitazioni proposte potrebbero richiedere più di 2 ore. Questo è voluto, per invogliarvi a completare l'esercitazione a casa e ripensare all'attività proposta.
2. In questa esercitazione, cercate di svolgere almeno i primi tre esercizi durante la lezione di laboratorio.
3. Gli esercizi asteriscati presentano difficoltà superiori rispetto a quelle degli esercizi non asteriscati proposti nel contesto di una esercitazione.
4. Se volete conservare i file creati durante le esercitazioni, è opportuno effettuare un backup di tali file ad esempio copiandoli su una chiavetta o inviandovi via e-mail.

## 2 Riepilogo comandi

<code>dir</code>	mostra il contenuto della cartella corrente
<code>cd <i>nomecartella</i></code>	naviga nella sottocartella <i>nomecartella</i>
<code>cd ..</code>	naviga nella cartella superiore
<code>md <i>nomecartella</i></code>	crea una sottocartella <i>nomecartella</i>
<code>gcc -ansi -Wall <i>n.c</i> -o <i>n.exe</i></code>	compila il file <i>n.c</i> generando il file <i>n.exe</i>
<code><i>n.exe</i></code>	esegue il programma <i>n.exe</i>

### 2.1 Esercizio 1

Scrivete un programma per la stampa di biglietti cinematografici. Una volta lanciato, il programma chiede all'utente il costo di ogni singolo biglietto (valore decimale), il nome del cliente e il numero di posti desiderati. Una volta che l'utente ha inserito i dati richiesti il programma aspetta una conferma, chiedendo

all'utente di scrivere un carattere ('S' o 'N'). Se l'utente non conferma ('N') il programma termina, altrimenti ('S') stampa un biglietto contenente il nome del cliente, il numero di posti richiesti e il costo del biglietto, come segue:

```
#####  
# Nome: Mario Rossi  
# Numero posti: 2  
# Prezzo: 22,50  
#####
```

Dopo la stampa il programma termina.

*Nota:* Quando leggete un numero tramite `scanf` il carattere di andata a capo (`\n`) rimane all'interno del buffer. Se dopo aver letto il numero leggete un singolo carattere (`scanf("%c", ...)`) verrà letto il simbolo di andata a capo e il programma non attenderà l'input dell'utente. Per ovviare a questo problema potete usare `scanf("\n%c", ...)`.

## 2.2 Esercizio 2

Modificate il programma realizzato per l'Esercizio 1 di modo che non termini dopo avere richiesto i dati (ed eventualmente stampato un biglietto) ma che continui a richiedere nuovi dati all'utente, utilizzando un ciclo infinito. Il ciclo deve terminare quando l'utente fornisce il carattere EOF, corrispondente a CTRL-Z da input.

*Nota:* Per terminare un programma potete anche premere CTRL-C. In questo caso, si interrompe l'esecuzione incondizionatamente.

## 2.3 Esercizio 3

1. Scrivere un programma che chiede all'utente di inserire un numero intero `n` e quindi legge `n` numeri interi in sequenza. Quindi, conta quante volte è stato inserito un numero pari e stampa a video tanti asterischi quanti sono stati i numeri pari inseriti. Utilizzare un ciclo `for` per realizzare l'iterazione.
2. Come sopra, ma utilizzare un ciclo `while` per realizzare l'iterazione.
3. Modificare uno dei due programmi sviluppati in risposta ai punti (1) e (2) precedenti affinché termini quando l'utente inserisce uno 0 (quindi, in questo caso, il numero `n` non deve essere letto. Utilizzare a questo proposito il costrutto di iterazione ritenuto più opportuno.

## 2.4 Esercizio 4

Scrivere un programma che chiede all'utente di inserire un numero intero `n`. Stampate quindi `n` linee, dove la prima contiene un asterisco, la seconda due e così via. In ogni riga, gli asterischi devono essere preceduti da tanti punti

quanti sono necessari per avere tutte righe della stessa lunghezza. Ad esempio, se l'utente inserisce il numero 5 il risultato deve essere il seguente:

```
....*
...**
..***
.****
*****
```

## 2.5 Esercizio 5\*

Scrivere un programma che chiede all'utente una data (nel formato giorno-mese-anno, es: 20/11/2011) e restituisce la data successiva. Considerate a questo proposito il diverso numero di giorni in ogni mese e gli anni bisestili. (Un anno è bisestile se è divisibile per 4 ma non per 400).

## 2.6 Esercizio 6\*

Modificate il programma realizzato per l'Esercizio 1 (oppure per l'Esercizio 2). Il nuovo programma NON dovrà chiedere il prezzo del biglietto ma l'età del cliente e l'ora corrente (utilizzate ad esempio 1200 per 12:00). Gli adulti (almeno 13 anni) pagano 8 euro fra le 12:00 e le 22:00, 5 euro prima delle 12:00 e 6 euro dopo le 22:00. I bambini (meno di 13 anni) pagano 5 euro fra le 12:00 e le 22:00, 3 euro prima delle 12:00 e 4 euro dopo le 22:00.

*Nota:* usate `if` innestati. Di solito è una buona idea separare la raccolta delle informazioni dalla loro elaborazione.