Esercizi

9.4 Scrivete un'istruzione **printf** o **scanf** per ognuna delle seguenti attività:

a) Visualizzate l'intero senza segno **40000** giustificato a sinistra in un campo di **15** cifre e con un minimo di **8** cifre.

b) Leggete un valore esadecimale nella variabile **hex**.

c) Visualizzate **200** con e senza segno.

d) Visualizzate **100** in formato esadecimale preceduto da **0x**.

e) Leggete dei caratteri nel vettore **s** finché non incontrate la lettera **p**.

f) Visualizzate **1,234** in un campo di **9** cifre preceduto da degli zeri.

g) Leggete un orario nel formato **hh:mm:ss** immagazzinando le sue parti nelle variabili intere **hour**, **minute** e **second**. Ignorate i due punti (:) inseriti nello stream di input. Utilizzate il carattere di soppressione dell'assegnamento.

h) Leggete dallo standard input una stringa nel formato **"characters"**. Immagazzinatela nel vettore di caratteri **s**. Eliminate le virgolette dallo stream di input.

i) Leggete un orario nel formato **hh:mm:ss** immagazzinando le sue parti nelle variabili intere **hour**, **minute** e **second**. Ignorate i due punti (:) inseriti nello stream di input. Non utilizzate il carattere di soppressione dell'assegnamento.

9.5 Mostrate che cosa sarà visualizzato da ognuna delle seguenti istruzioni. Nel caso che un'istruzione non sia corretta, indicatene il motivo.

a) `printf("%-10d\n", 10000);`

b) `printf("%c\n", "This is a string");`

c) `printf("%*.*lf\n", 8, 3, 1024.987654);`

d) `printf("%#o\n%#X\n%#e\n", 17, 17, 1008.83689);`

e) `printf("% ld\n%+ld\n", 1000000, 1000000);`

f) `printf("%10.2E\n", 444.93738);`

g) `printf("%19.2g\n", 444.93738);`

h) `printf("%d\n", 10.987);`

9.6 Trovate l'errore (o gli errori) in ognuno dei seguenti segmenti di programma. Per ognuno di essi, spiegate come possa essere corretto.

a) `printf("%s\n", 'Happy Birthday');`

b) `printf("%c\n", 'Hello');`

c) `printf("%c\n", "This is a string");`

d) L'istruzione successiva dovrebbe visualizzare **"Bon Voyage"**.

```
printf(""%s"", "Bon Voyage");
```

e)
```
char day[] = «Sunday»;
printf("%s\n", day[3]);
```

f) `printf('Enter your name: ');`

g) `printf(%f, 123.456);`

h) L'istruzione successiva dovrebbe visualizzare i caratteri **'O'** e **'K'**.

```
printf("%s%s\n", 'O', 'K');
```

i)
```
char s[10];
scanf("%c", s[7]);
```

9.7 Scrivete un programma che inizializzi i 10 elementi del vettore **number** con degli interi casuali compresi tra 1 e 1000. Visualizzate ogni valore e il totale progressivo del numero di caratteri visualizzati. Utilizzate la specifica di conversione **%n** per determinare il numero di caratteri inviati in output per

ogni valore. Visualizzate il totale dei caratteri inviati in output per tutti i valori, includendo anche quello corrente, ogni volta che sarà visualizzato. L'output dovrà avere il seguente formato:

```
Value     Total characters
342       3
1000      7
963       10
6         11
etc.
```

9.8 Scrivete un programma che verifichi la differenza tra gli indicatori di conversione **%d** e **%i** quando sono utilizzati nelle istruzioni **scanf**. Utilizzate le istruzioni

```
scanf("%i%d", &x, &y);
printf("%d %d\n", x, y);
```

per prendere in input e visualizzare i valori. Verificate il funzionamento del programma con i seguenti gruppi di dati:

```
  10    10
 -10   -10
 010   010
0x10  0x10
```

9.9 Scrivete un programma che visualizzi i valori di un puntatore utilizzando tutti gli indicatori di conversione per gli interi e la specifica **%p**. Quale visualizzerà dei valori strani? Quale provocherà degli errori? Sul vostro sistema, in quale formato visualizzerà l'indirizzo la specifica di conversione **%p**?

9.10 Scrivete un programma che verifichi i risultati ottenuti visualizzando il valore intero **12345** e quello in virgola mobile **1,2345**, in campi con varie dimensioni. Che cosa succederà quando i valori saranno visualizzati in campi con una dimensione inferiore a quelle dei valori?

9.11 Scrivete un programma che visualizzi il valore **100,453627** arrotondato all'intero più vicino e al numero più vicino a meno di un decimo, di un centesimo, di un millesimo e di un decimo di millesimo.

9.12 Scrivete un programma che prenda in input dalla tastiera una stringa e ne determini la lunghezza. Visualizzate la stringa utilizzando come dimensione di campo il doppio della sua lunghezza.

9.13 Scrivete un programma che converta degli interi, corrispondenti a temperature Fahrenheit comprese tra **0** e **212**, nelle equivalenti Celsius espresse in numeri a virgola mobile con **3** cifre di precisione. Utilizzate la formula

```
celsius = 5.0 / 9.0 * (fahrenheit - 32);
```

per eseguire il calcolo. L'output dovrà essere visualizzato giustificato a destra all'interno di due colonne, ognuna di 10 caratteri, e le temperature Celsius dovranno essere precedute da un segno sia per i valori positivi, sia per quelli negativi.

9.14 Scrivete un programma che verifichi tutte le sequence di escape della Figura 9.16. Visualizzate un carattere prima e dopo le sequenze di escape che spostano il cursore, per rendere evidente lo spostamento del cursore.

9.15 Scrivete un programma che determini se il **?** possa essere visualizzato come carattere letterale, nella stringa di controllo del formato di **printf**, invece che con la sequenza di escape **\?**.

9.16 Scrivete un programma che prenda in input il valore **437** utilizzando con **scanf** ognuno degli indicatori di conversione per gli interi. Visualizzate ogni valore ricevuto in input utilizzando tutti gli indicatori di conversione per gli interi.

9.17 Scrivete un programma che utilizzi gli indicatori di conversione **e**, **f** e **g** per prendere in input il valore **1,2345**. Visualizzate i valori di ogni variabile per dimostrare che i suddetti indicatori di conversione potranno essere utilizzati per prendere in input lo stesso valore.

9.18 In alcuni linguaggi di programmazione, le stringhe devono essere immesse delimitandole con apici singoli o virgolette. Scrivete un programma che legga le tre stringhe **suzy**, **"suzy"** e **'suzy'**. Il C ignora gli apici singoli e le virgolette o li considera parti integranti della stringa?

9.19 Scrivete un programma che determini se, utilizzando l'indicatore di conversione **%c** nella stringa di controllo del formato di un'istruzione **printf**, il **?** possa essere visualizzato con la costante di carattere **'?'** invece che con la corrispondente sequenza di escape.

9.20 Scrivete un programma che utilizzi l'indicatore di conversione **g** per inviare in output il valore **9876,12345**. Visualizzate il valore con delle precisioni comprese tra **1** e 9.