Esercizio 1

Si convertano i numeri decimali 11 e -5 in numeri binari di 5 bit (usando il complemento a 2 per i numeri negativi) e si esegua l'operazione 11 - 5 in binario. Si converta poi il risultato in decimale verificandone la correttezza.

Correzione

Il numero 11_{10} si scrive 01011_2, mentre il numero -5_{10} si scrive 11011_2 in complemento a 2 (infatti il numero 5_{10} si scrive 00101_2, e dopo aver negato tutti i bit diventa 11010_2 a cui va sommato 1). La sottrazione tra 11_{10} e 5_{10} equivale quindi alla somma tra 11_{10} e -5_{10}, che fa 00110_2 e che equivale infatti a 6_{10} ($2^1 + 2^2$).

Esercizio 2

Come è codificata l'informazione, e.g., testo, immagini, suoni, nei calcolatori? (si dia una spiegazione intuitiva di come questa rappresentazione possa essere implementata con dispositivi elettronici).

Correzione

Occorreva illustrare brevemente il funzionamento alla base del codice binario, grazie al quale si possono rappresentare le varie informazioni tramite sequenza di bit. In particolare per il testo su usa ASCII e UNICODE, per immagini Bitmap, griglie di pixel, RGB, etc. Per i suoni si effettua un campionamento e si converte l'analogico in digitale. Volendo si poteva anche citare la differente rappresentazione dell'informazione (corrente si/no) dei CD e DVD.

Esercizio 3

Leggere attentamente il seguente listato di codice. Il main presenta svariati errori; dopo averli individuati e motivati, correggerli riscrivendo la funzione main e illustrare cosa stampa il programma dopo aver letto la stringa `"Prova"`.

```
1  #include <stdio.h>
2
3  char str[20];
4
5  int* fun(char *s)
6  {
7    int* n, i=0;
8    for(i=0;s[i]!='\0';i++) ;
9    *n=i;
10   for(i=0;i<*n/2;i++)
11   {
12   s[*n-i-1]=s[i];
13   s[i]=s[*n-i-1];
14   }
15   return n;
16 }
17
18 int main(void)
19 {
20      scanf("%19s",str);
21      len = fun(str);
22      printf('La stringa modificata e' %c ed e' lunga %c.\n',len,str);
23      return(0);
24 }
```

Correzione

Anzitto la variabile `str` e' dichiarata correttamente. E' una variabile globale quindi viene vista nel main senza che la si ridichiari o che se ne sposti la dichiarazione a riga 20. Quindi non era un errore (da molti segnalato).

Anche a riga 20 non ci sono errori. La scanf richiede un indirizzo e il nome della stringa e' gia' un puntatore. Inoltre nel template possiamo mettere `"%19s"` senza problemi, anzi, ci assicura che non sforeremo la memoria allocata per str.

A riga 21 `len` non e' dichiarata. `fun` ritorna un puntaore quindi occorre dichiararla come puntatore ad intero, oppure come intero ma in tal caso il valore di ritorno va dereferenziato.

A riga 22 il template e' sbagliato: ci vanno i doppi apici, i `%c` vanno sostituiti con `%s` e `%d` o `%i` e anche l'ordine dei parametri e' invertito.

Il main correttto e' il seguente e dopo aver letto la stringa `"Prova"` stampa
`La stringa modificata e' ProrP ed e' lunga 5.`

```
int main(void)
{
  int len;
  scanf("%19s",str);
  len = *fun(str);
  printf("La stringa modificata e' %s ed e' lunga %d.\n",str,len);
  return(0);
}
```