

## Esercizio 1

Dato un numero binario  $X$  di 3 cifre  $X_1$ ,  $X_2$  e  $X_3$ , si consideri la funzione booleana  $f(X_1, X_2, X_3)$  che ritorna 1 se il numero di cifre uguali ad 1 è pari e ritorna 0 se il numero di cifre uguali ad 1 è dispari. Si costruisca la tabella di verità per la funzione  $f()$  e si esprima la funzione in forma canonica.

### Correzione

Occorreva costruire una tabella di verità con 8 righe ( $2^3$  perché ci sono 3 variabili booleane). Il valore di verità 1 andava assegnato alle seguenti combinazioni di  $X_1, X_2, X_3$ : 000 011 101 110. Quindi la forma canonica di  $f(X_1, X_2, X_3)$  risultava essere:  $\bar{X}_1\bar{X}_2\bar{X}_3 + \bar{X}_1X_2X_3 + X_1\bar{X}_2X_3 + X_1X_2\bar{X}_3$

Alcuni hanno omissso la prima combinazione, cioè  $\bar{X}_1\bar{X}_2\bar{X}_3$  pensando che il numero di 1, essendo 0, non fosse positivo. Ma mettendo 0 avete supposto che 0 fosse un numero dispari, dunque, sbagliando.

## Esercizio 2

Quali sono le fasi di progettazione del software? (si dia una breve spiegazione per ogni fase).

### Correzione

Il ciclo di vita di un software è diviso in 5 fasi principali: analisi dei requisiti, progettazione, codifica, testing e manutenzione. Per ogni fase trovate la spiegazione sulle slides (o su wikipedia o qualsiasi libro di informatica).

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

### Correzione

Anzitutto 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 19. Quindi non era un errore (da molti segnalato).

A riga 20 la scanf richiede un indirizzo e il nome della stringa e' gia' un puntatore, non serve la **&**. Inoltre nel template vanno i doppi apici, non l'apice singolo.

A riga 21 fun e' una procedura e non ha valore di ritorno quindi. Il parametro len va passato come riferimento, usando **&**. Si sarebbe potuto anche usare **len**. da solo ma allora bisognava dichiararlo **int \*** ma in quel caso bisognava allorargli anche dello spazio.

A riga 22 il template e' sbagliato mentre a riga 23 exit e' una funzione quindi il parametro 0 va tra parentesi.

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

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