

**Esercizio 1**

Si discutano in dettaglio le varie operazioni di casting che possono avvenire in un programma C.

**Esercizio 2**

Spiegare quale sia il numero minimo di bit necessari a rappresentare i numeri  $-10_{10}$  e  $-8_{10}$  e la loro somma, usando la rappresentazione binaria in complemento a 2.

### Esercizio 3

Leggere attentamente il seguente listato di codice. Il main presenta svariati errori; dopo averli individuati e motivati, correggerli e illustrare cosa stampa il programma dopo aver letto: "1 13 131 1 3 1".

```
1 #include <stdio.h>
2
3 int fun(int *a, char *b)
4 {
5     int i=0,j=0;
6     while (a[i])
7     {
8         if ((a[i]>0)&&(a[i]<26))
9         {
10             b[j]=a[i]+'a'-1;
11             j++;
12         }
13         i++;
14     }
15     b[j]='\0';
16     return j;
17 }
18
19 int main(void)
20 {
21     char str[20];
22     int val,k=0;
23
24     while (1)
25     {
26         scanf( '%d' ,val );
27         if (val<=0)
28             continue;
29         num[k]=val;
30         k++;
31     }
32     if (k<=0)
33         printf("Stringa vuota\n");
34     else
35         printf("La stringa ottenuta e' %s ed e' lunga %d",str,fun(num,str));
36     exit(0);
37 }
```

### Correzione

Anzitutto a riga 23 la variabile **num** non e' dichiarata; E' un array di numeri: **static int num[20];**

A riga 26 nel template vanno i doppi apici, non l'apice singolo e l'operatore **&**. **scanf("%d",&val);**

A riga 28 il **continue** va sostituito con **break**;

La stringa **str** viene costruita a partire da **num** nella funzione **fun**. Per correggere l'errore di riga 23 bisogna spezzare l'output della printf nel seguente modo:

```
k=fun(num,str);    printf("La stringa ottenuta e' %s ed e' lunga %d",str,k);
```

Nell'input affinche finisca il ciclo infinito bisogna dargli un numero negativo. Con input 1 12 131 1 3 1 seguito da -1 stampa:

La stringa ottenuta e' amaca ed e' lunga 5.