

## Rappresentazione dei valori booleani in C

**Premessa fondamentale:** nel linguaggio C non esiste il tipo dati booleano.

Pertanto i valori possibili per questo tipo di dato (VERO e FALSO) devono essere “simulati” con il tipo dati primitivo **int**.

**N.B. Di norma in C si associa al valore di verità FALSO il valore numerico intero 0 mentre al valore di verità VERO un valore numerico diverso da 0 (ad esempio proprio 1)**

**Esempio:** sia dato il seguente frammento di algoritmo scritto in pseudocodifica da implementare successivamente usando il linguaggio C:

```
ALGORITMO A
PROCEDURA main( )
INIZIO
risposta: BOOL
a: INT
risposta ← VERO
SE (risposta = VERO)
  ALLORA
    a ← a + 1
  ALTRIMENTI
    a ← a - 1
  risposta ← FALSO
FINE SE
Scrivi (risposta)
FINE
```

### 1) Valori booleani rappresentati direttamente attraverso il tipo di dati int

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  int risposta;
  int a;

  risposta = 1;
  if (risposta == 1)
  {
    a = a + 1;
  }
  else
  {
    a = a - 1;
    risposta = 0;
  }

  printf ("%d", risposta);
  system("PAUSE");
  return 0;
}
```

**N.B. ovviamente a video verrà mostrato il valore numerico 1**

**2) Valori booleani rappresentati attraverso costanti appartenenti al tipo di dati int**

```
#include <stdio.h>
#include <stdlib.h>

#define FALSO 0
#define VERO 1

int main(int argc, char *argv[])
{
    int risposta;
    int a;

    risposta = VERO;
    if (risposta == VERO)
    {
        a = a + 1;
    }
    else
    {
        a = a -1;
        risposta = FALSO;
    }

    printf ("%d", risposta);
    system("PAUSE");
    return 0;
}
```

**N.B. ovviamente a video verrà sempre mostrato il valore numerico 1**

**3) Valori booleani rappresentati attraverso l'uso dell'istruzione "typedef" e del costrutto "enum"**

```
#include <stdio.h>
#include <stdlib.h>

typedef enum {FALSO=0, VERO=1} BOOL;

int main(int argc, char *argv[])
{
    BOOL risposta;
    int a;

    risposta = VERO;
    if (risposta == VERO)
    {
        a = a + 1;
    }
    else
    {
        a = a -1;
        risposta = FALSO;
    }

    printf ("%d", risposta);
    system("PAUSE");
    return 0;
}
```

**N.B. anche in quest'ultimo caso a video verrà mostrato il valore numerico 1**