

# Lezione 18

30 novembre–1 dicembre 1999

## Argomenti trattati

- Esempi di procedure ricorsive.
- Esercizi di ripasso.

## 18.1 Esempio: da notazione decimale a notazione binaria

Presentiamo un programma che leggendo da **input** un numero intero espresso in notazione decimale, scrive in **output** la notazione binaria corrispondente.

Il programma, dopo avere letto il numero da **input**, richiama una procedura che si occupa di determinare e di scrivere in **output** la stringa binaria corrispondente al numero dato. Tale procedura, di nome **scrivibinario**, riceve come parametro il valore da convertire. Dunque la sua intestazione sarà:

```
PROCEDURE scrivibinario (x: integer);
```

Possiamo subito descrivere il programma principale:

```
PROGRAM conversione (input, output);
```

```
{converte un intero da notazione decimale a notazione binaria}
```

```
VAR
```

```
  numero: integer;
```

```
PROCEDURE scrivibinario (x: integer);
```

```
  {scrive in output, in notazione binaria, il valore del parametro}
```

```
  {versione iterativa}
```

```
  ...dichiarazioni e codice della PROCEDURE scrivibinario...
```

```
BEGIN {conversione}
```

```
  readln(numero);
```

---

©1999 Giovanni Pighizzini

Il contenuto di queste pagine è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle pagine sono di proprietà dell'autore. Le pagine possono essere riprodotte ed utilizzate liberamente dagli studenti, dagli istituti di ricerca, scolastici ed universitari afferenti ai Ministeri della Pubblica Istruzione e dell'Università e della Ricerca Scientifica e Tecnologica per scopi istituzionali, non a fine di lucro. Ogni altro utilizzo o riproduzione (ivi incluse, ma non limitatamente a, le riproduzioni a mezzo stampa, su supporti magnetici o su reti di calcolatori) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte dell'autore.

L'informazione contenuta in queste pagine è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, ecc.

L'informazione contenuta in queste pagine è soggetta a cambiamenti senza preavviso. L'autore non si assume alcuna responsabilità per il contenuto di queste pagine (ivi incluse, ma non limitatamente a, la correttezza, completezza, applicabilità ed aggiornamento dell'informazione). In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste pagine. In ogni caso questa nota di copyright non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

```

    scrivibinario(numero);
    writeln
END. {conversione}

```

Sviluppiamo ora la **PROCEDURE** `scrivibinario`.

La conversione di un numero, da binario a decimale, può essere effettuata dividendo iterativamente il numero per 2 sino ad ottenere zero. La sequenza dei resti, letta al contrario, fornisce la stringa binaria corrispondente. Ad esempio, se il numero dato è 11, si possono effettuare le seguenti divisioni:

- 11 DIV 2 = 5 con resto 1;
- 5 DIV 2 = 2 con resto 1;
- 2 DIV 2 = 1 con resto 0;
- 1 DIV 2 = 0 con resto 1

Leggendo al contrario la sequenza dei resti, si ottiene la rappresentazione binaria, in questo caso 1011.

Una prima soluzione, basata su questa tecnica, consiste nell'effettuare le divisioni all'interno di un ciclo, memorizzando i resti in un array. Leggendo l'array dal fondo si ricava la notazione binaria.

Definiamo dunque un array, i cui elementi possono essere i caratteri '0' e '1'. Usiamo come indici gli interi a partire da 1, fissando, con una costante, la lunghezza massima dell'array. Introduciamo, inoltre, una variabile di nome `pos`, che usiamo di volta in volta per sapere fino a che punto è stato riempito l'array. Dunque, forniamo le seguenti dichiarazioni:

```

CONST
    max = 10;

TYPE
    cifrabinaria = '0'..'1';

VAR
    a: ARRAY[1..max] OF cifrabinaria;
    pos: 0..max;

```

La procedura viene suddivisa in due fasi: nella prima si riempie l'array `a` con le cifre binarie corrispondenti al numero dato, nella seconda si scrive il risultato in `output`.

Possiamo basare la prima fase sul seguente ciclo:

```

WHILE x e' diverso da zero DO
    dividi x per 2 ponendo la cifra corrispondente al
    resto nella prima posizione libera di a

```

Utilizzando `pos` per indicare la prima posizione libera di `a`, questa fase può essere scritta in Pascal come:

```

pos := 0;
WHILE x <> 0 DO
    BEGIN
        pos := pos + 1;
        IF x MOD 2 = 0 THEN
            a[pos] := '0'
        ELSE

```

```

    a[pos] := '1';
    x := x DIV 2
END

```

A questo punto, la scrittura può essere effettuata scandendo l'array all'indietro, a partire dalla posizione `pos`, con un ciclo `FOR`. Come variabile di controllo del ciclo possiamo utilizzare sempre `pos`:

```

FOR pos := pos DOWNTO 1 DO
    write(a[pos]);
writeln

```

Il codice completo della procedura così costruita è:

```

PROCEDURE scrivibinario (x: integer);

    {scrive in output, in notazione binaria, il valore del parametro}
    {versione iterativa}

    CONST
        max = 10;

    TYPE
        cifrabinaria = '0'..'1';

    VAR
        a: ARRAY[1..max] OF cifrabinaria;
        pos: 0..max;

    BEGIN {scrivibinario}
        {fase di conversione}
        pos := 0;
        WHILE x <> 0 DO
            BEGIN
                pos := pos + 1;
                IF x MOD 2 = 0 THEN
                    a[pos] := '0'
                ELSE
                    a[pos] := '1';
                x := x DIV 2
            END; {while}

            {fase di scrittura}
            FOR pos := pos DOWNTO 1 DO
                write(a[pos])
            END; {scrivibinario}
    END;

```

Si osservi che se `x` inizialmente contiene zero, la procedura precedente non stampa nulla, mentre dovrebbe stampare la cifra 0. Questo è dovuto al fatto che, in questo caso, non viene inserito alcun elemento all'interno dell'array `a`. Questo problema può essere facilmente risolto sostituendo il ciclo `WHILE` con un ciclo `REPEAT`:

```

PROCEDURE scrivibinario (x: integer);

```

```

{scrive in output, in notazione binaria, il valore del parametro}
{versione iterativa}

CONST
    max = 10;

TYPE
    cifrabinaria = '0'..'1';

VAR
    a: ARRAY[1..max] OF cifrabinaria;
    pos: 0..max;

BEGIN {scrivibinario}
    {fase di conversione}
    pos := 0;
    REPEAT
        pos := pos + 1;
        IF x MOD 2 = 0 THEN
            a[pos] := '0'
        ELSE
            a[pos] := '1';
        x := x DIV 2
    UNTIL x = 0;

    {fase di scrittura}
    FOR pos := pos DOWNTO 1 DO
        write(a[pos])
    END; {scrivibinario}

```

La procedura sviluppata richiede l'utilizzo di un array, la cui lunghezza potrebbe risultare insufficiente per il numero fornito in ingresso. Occorrerebbe introdurre nel ciclo un controllo per trattare il caso in cui si superi la dimensione dell'array.

Sviluppiamo ora seconda versione della procedura in cui, sfruttando la ricorsione, evitiamo di introdurre l'array.

Osserviamo che se  $x$  contiene 0 o 1, la rappresentazione binaria è ottenibile immediatamente (caso base). Altrimenti, l'*ultima cifra* della rappresentazione binaria di  $x$  corrisponde al resto della divisione di  $x$  per 2, mentre le cifre precedenti costituiscono la rappresentazione binaria del quoziente della divisione di  $x$  per 2. Ad esempio, la rappresentazione binaria di 11 si ottiene concatenando alla rappresentazione binaria di 5 ( $11 \text{ DIV } 2$ ), la cifra 1 (corrispondente a  $11 \text{ MOD } 2$ ). A sua volta, la rappresentazione binaria di 5 può essere ottenuta ricorsivamente utilizzando la stessa regola. Applicando la regola ricorsiva appena stabilita, possiamo costruire una procedura secondo il seguente schema:

```

IF (x = 0) OR (x = 1)
THEN
    scrivi la cifra binaria corrispondente a x
ELSE
    BEGIN
        scrivi la rappresentazione binaria di x DIV 2
        scrivi la cifra corrispondente a x MOD 2
    END

```

L'istruzione `scrivi la rappresentazione binaria di x DIV 2` viene realizzata mediante una chiamata ricorsiva. Il codice della procedura costruita secondo questo schema è:

```
PROCEDURE scrivibinario (x: integer);

  {scrive in output, in notazione binaria, il valore del parametro}
  {versione ricorsiva}

BEGIN {scrivibinario}
  IF x = 0 THEN
    write('0')
  ELSE IF x = 1 THEN
    write('1')
  ELSE
    BEGIN
      scrivibinario(x DIV 2);
      IF x MOD 2 = 0 THEN
        write('0')
      ELSE
        write('1')
    END
  END; {scrivibinario}
```

## Esercizi

1. Si simuli l'esecuzione della procedura `scrivibinario`, nella versione ricorsiva, evidenziando l'andamento dello stack. In particolare, si osservi che il ruolo svolto dall'array, utilizzato nella versione iterativa, è svolto, in questo caso, dallo stack della macchina Pascal. Si noti inoltre che, nella soluzione ricorsiva, è fondamentale l'utilizzo del passaggio *per valore*.
2. Cosa succede eseguendo il programma di conversione (sia nella versione iterativa, che in quella ricorsiva) su numeri negativi?