

# Steppin' Out

## PREMESSA:

Si faccia riferimento alla Guida. Ricordiamo che l'operatore `==` indica il test di uguaglianza. Non va confuso con l'assegnamento, che associa un valore ad una variabile. Dunque `A = 3` e `B = 4` sono due assegnamenti. Se, dopo tali istruzioni, valutiamo l'espressione `A == B`, questa sarà falsa, poiché non è vero che 3 è uguale a 4. Se poi modificheremo il valore di B con un nuovo assegnamento `B=3` allora, dopo tale istruzione, l'espressione `A == B` sarà vera, poiché sia A che B valgono 3.

Ricordiamo che il simbolo `^` indica l'elevamento a potenza. Dunque `2 ^ 3` vale 8, `3 ^ 2` vale 9, e così via.

All'inizio dell'esecuzione di una procedura, le variabili dichiarate di tipo integer hanno valore 0 e le variabili dichiarate di tipo string contengono la stringa vuota.

## PROBLEMA:

Sia data la seguente procedura

```
procedure STREETSCIENCE;
  variables A,B,C,D integer;
  read A,B,C,D;
  A = B + C * (A ^ 2);
  if A < 40 then {
    D = 100 - 1;
    D = D * 2;
  }; endif;
  if A > 100 then {
    A = (C ^ 2) + 2 - 10;
  }; endif;
  if A == 10 then {
    for K from 1 to 4 step 1 do;
      A = (A - 1) * 2;
      B = A;
      if B < 50 then {
        C = K * (8 / 2);
      }; else {
        C = 180 / K;
      }; endif;
    endfor;
  }; endif;
  write A, C, D;
end procedure;
```

La procedura viene eseguita per **tre volte**, variando i valori di input:

1. per la prima esecuzione, i valori di input per A, B, C e D sono (rispettivamente) 2, 4, 6, 7;
2. per la seconda esecuzione, i valori di input per A, B, C e D sono (rispettivamente) 4, 6, 10, 3;
3. per la terza esecuzione, i valori di input per A, B, C e D sono (rispettivamente) 1, 7, 3, 2.

**Calcolare e indicare nella tabella sottostante** i valori di output, nella forma [*valore di output di A, valore di output di C, valore di output di D*] (es. [3,4,5] significa che, al termine dell'esecuzione, A vale 3, B vale 4 e C vale 5).

<i>I esecuzione</i>	
<i>II esecuzione</i>	
<i>III esecuzione</i>	