

Programmazione I

Compitino del 2 dicembre 1999

1. Si considerino le seguenti definizioni di tipo:

```
TYPE lettera = 'A'..'Z';
   cifra = '0'..'9';
   indice = 1..10;
   a = RECORD
       tab: ARRAY[lettera, indice] OF cifra;
       ins: SET OF cifra
   END;
```

Scrivere in Pascal una **FUNCTION** che riceva un parametro **x** di tipo **a** e restituisca, come risultato, il numero di elementi dell'array memorizzato nel campo **tab** di **x**, che contengono valori appartenenti all'insieme memorizzato nel campo **ins** di **x**.

2. Per ognuna delle seguenti linee di codice individuare delle dichiarazioni di variabile ed eventualmente di tipo, nonché eventuali intestazioni di procedura o funzione, in modo che le istruzioni che vi appaiono risultino corrette dal punto di vista della compatibilità dei tipi. Se ciò non fosse possibile, spiegare il motivo.

- `a := -b[pred(a)]`
- `a := -b{pred(a)}`
- `a := -b(pred(a))`

3. Per ogni blocco di codice del seguente programma indicare i nomi delle variabili che possono essere utilizzate e delle procedure che possono essere richiamate, precisando, per ciascun nome, il blocco in cui è stata effettuata la dichiarazione.

```
PROGRAM p;
  VAR
    d, b: char;

  PROCEDURE c;
    VAR
      b: integer;
  BEGIN {c}
    ...
  END; {c}

  PROCEDURE a;
    VAR
      d, e: integer;

    PROCEDURE b;
      VAR
        c, f: integer;
    BEGIN {b}
      ...
    END; {b}

  BEGIN {a}
    ...
  END; {a}

BEGIN {p}
  ...
END. {p}
```

4. Per ognuno dei seguenti programmi scrivere l'output prodotto su input 5.

```
PROGRAM p1 (input, output);
```

```
VAR
```

```
  x, y: integer;
```

```
PROCEDURE a;
```

```
  VAR y: integer;
```

```
BEGIN {a}
```

```
  y := 3 * x;
```

```
  x := y - 1
```

```
END; {a}
```

```
PROCEDURE b;
```

```
  VAR x: integer;
```

```
BEGIN {b}
```

```
  x := y - 2;
```

```
  a
```

```
END; {b}
```

```
BEGIN {p1}
```

```
  readln(x);
```

```
  y := x - 1;
```

```
  b;
```

```
  writeln(x, y)
```

```
END. {p1}
```

```
PROGRAM p2 (input, output);
```

```
VAR
```

```
  x, y: integer;
```

```
FUNCTION f (n: integer): integer;
```

```
BEGIN {f}
```

```
  IF (n MOD 3 = 0) OR (n <= 1) THEN
```

```
    f := n
```

```
  ELSE IF n MOD 2 = 0 THEN
```

```
    f := 3 * f(n DIV 2)
```

```
  ELSE
```

```
    f := 2 * f(n - 1) + 2 * f(n - 2)
```

```
END; {f}
```

```
BEGIN {p2}
```

```
  readln(x);
```

```
  y := f(x);
```

```
  writeln(y)
```

```
END. {p2}
```

```
PROGRAM p3 (input, output);
```

```
VAR
```

```
  x, y: integer;
```

```
PROCEDURE a (v: integer;
```

```
  VAR w: integer);
```

```

BEGIN
  w := 3 * (v + 2);
  v := w - v
END;

```

```

BEGIN {p3}
  readln(x);
  a(x, y);
  writeln(x, y)
END. {p3}

```

5. Indicare come potrebbe essere strutturato un programma **p** che contenga tre procedure **a**, **b** e **c**, e nel quale non sia utilizzata la direttiva **forward**, in modo tale che:

- la procedura **b** sia in grado di richiamare le altre due procedure, ma non sia richiamabile dal programma principale; il programma principale sia in grado di richiamare la procedura **a**.
- il programma principale possa richiamare la procedura **b** e la procedura **a**, la procedura **b** possa richiamare la procedura **c**, ma non la procedura **a**, la procedura **a** possa richiamare tutte le procedure.

6. Si consideri la funzione f sugli interi definita come:

$$f(x) = \begin{cases} 1 & \text{se } x \leq 1 \\ 2 * f(x - 1) + f(x - 2) & \text{altrimenti.} \end{cases}$$

Scrivere in Pascal una **FUNCTION** **f** ricorsiva per il calcolo di f .

Indicare il valore di $f(4)$.