1.

a. Programmet skriver ut `1 1 1 1 1`.

b. Parametern `n` måste vara en `VAR`-parameter för att argumentvariablen `x` ska dubblas.

```
PROCEDURE dubbla(VAR n:INTEGER);
BEGIN
  n:=2*n
END dubbla;
```

2.

a.

```
PROCEDURE ReadLine(VAR s:ARRAY OF CHAR);
  VAR i:CARDINAL; c:CHAR;
BEGIN
  i:=0;
  LOOP
    Read(c);
    IF NOT Done() OR (c=12C) THEN EXIT END;
    IF i<=HIGH(s) THEN s[i]:=c END;
    INC(i)
  END;
  IF i<=HIGH(s) THEN s[i]:=0C END
END ReadLine;
```

b.

```
PROCEDURE ReadCard(VAR n:CARDINAL);
  VAR c:CHAR;
BEGIN
  n:=0;
  LOOP
    Read(c);
    IF NOT Done() OR (c<"0") OR (c>"9") THEN EXIT END;
    n:=10*n+ORD(c)-ORD("0")
  END
END ReadCard;
```

3.

a.

```
TYPE Month = (Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec);
     Day = [1..31];
     Date = RECORD
         year:INTEGER;
         month:Month;
         day:Day
       END;
```

b.

```
PROCEDURE WriteMonth(m:Month);
BEGIN
  CASE m OF
   Jan: WriteString("Januari")
  |Feb: WriteString("Februari")
  |Mar: WriteString("Mars")
  |Apr: WriteString("April")
  |May: WriteString("Maj")
```

```
    |Jun: WriteString("Juni")
    |Jul: WriteString("Juli")
    |Aug: WriteString("Augusti")
    |Sep: WriteString("September")
    |Oct: WriteString("Oktober")
    |Nov: WriteString("November")
    |Dec: WriteString("December")
  END
END WriteMonth;

PROCEDURE WriteDate(d:Date);
BEGIN
  WriteCard(d.day,1); Write(" ");
  WriteMonth(d.month); Write(" ");
  WriteInt(d.year,1)
END WriteDate;
```

4.

a.

```
PROCEDURE swap(VAR a,b:CHAR);
(* En hjälpprocedur för att byta plats på två tecken *)
  VAR t:CHAR;
BEGIN
  t:=a; a:=b; b:=t
END swap;

PROCEDURE RevString(VAR s:ARRAY OF CHAR);
  VAR n,i:INTEGER;
BEGIN
  n:=Length(s);
  FOR i:=0 TO n DIV 2 -1 DO
    swap(s[i],s[n-1-i])
  END
END RevString;
```

b.

```
PROCEDURE fac(n:CARDINAL):CARDINAL;
BEGIN
  IF n<2 THEN RETURN 1
  ELSE RETURN n*fac(n-1)
  END
END fac;
```

5.

```
MODULE histo;

FROM InOut IMPORT Read,Done,Write,WriteString,WriteCard,WriteLn;

TYPE Bokstav = ["A".."Z"];
     Histogram = ARRAY Bokstav OF CARDINAL;

VAR b:Bokstav;
    h:Histogram;
    n:CARDINAL;
    c:CHAR;

PROCEDURE stapel(n:CARDINAL);
  VAR i:CARDINAL;
BEGIN
  FOR i:=1 TO n DO
    Write("*")
  END;
  WriteLn
END stapel;

BEGIN
  FOR b:=MIN(Bokstav) TO MAX(Bokstav) DO
    h[b]:=0
  END;

  LOOP
    Read(c);
    IF NOT Done() THEN EXIT END;
    CASE CAP(c) OF
      "A".."Z": INC(h[CAP(c)]);
    ELSE
      (* inget att göra *)
    END
  END;

  FOR b:=MIN(Bokstav) TO MAX(Bokstav) DO
    IF h[b]>0 THEN
      Write(b); Write(" "); stapel(h[b])
    END
  END
END histo.
```

6.

a.

```
IMPLEMENTATION MODULE StringBag;

FROM Storage IMPORT ALLOCATE;
FROM InOut IMPORT WriteString,WriteCard,WriteLn;
FROM Strings IMPORT compare;

TYPE StringBag = POINTER TO Node;
     Node = RECORD
         str: String;(* en sträng *)
         cnt: CARDINAL;(* antal kopior av denna sträng *)
         left, right:StringBag
       END;

PROCEDURE NewStringBag(VAR t:StringBag);
```

```
BEGIN
  t:=NIL
END NewStringBag;

PROCEDURE NewLeaf(s:String):StringBag;
  VAR t:StringBag;
BEGIN
  NEW(t);
  t^.str:=s;
  t^.cnt:=1;
  t^.left:=NIL;
  t^.right:=NIL;
  RETURN t
END NewLeaf;

PROCEDURE InsertString(s:String; VAR t:StringBag);
BEGIN
  IF t=NIL THEN
    t:=NewLeaf(s)
  ELSE
    CASE compare(s,t^.str) OF
      -1: InsertString(s,t^.left)
    |  1: InsertString(s,t^.right)
    |  0: INC(t^.cnt)
    END
  END
END InsertString;

PROCEDURE CountString(s:String; t:StringBag):CARDINAL;
BEGIN
  (* behövs ej *)
END CountString;

PROCEDURE WriteStringBag(t:StringBag);
BEGIN
  IF t#NIL THEN
    WriteStringBag(t^.left);
    WriteString(t^.str); WriteCard(t^.cnt,5); WriteLn;
    WriteStringBag(t^.right);
  END
END WriteStringBag;

END StringBag.
```

b.

```
MODULE wordhisto;

FROM StringBag IMPORT
String,StringBag,NewStringBag,InsertString,WriteStringBag;
FROM InOut IMPORT ReadString,Done;

VAR wb:StringBag;
    s:String;

BEGIN
  NewStringBag(wb);
  LOOP
    ReadString(s);
    IF NOT Done() THEN EXIT END;
    InsertString(s,wb)
  END;
  WriteStringBag(wb)
END wordhisto.
```