```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '82 }
{ -- PASCAL PROGRAM SOLUTIONS }


{1.1}
program One1T82;
{ -- This program will allow a user to guess a generated #. }
  var
    X, I, G: Byte;

begin
  Randomize;
  X := Random(100) + 1;  I := 1;
  while (I <= 7) and (G <> X) do begin
    Write ('I AM THINKING OF A NUMBER. WHAT IS IT? ');
    Readln (G);
    if G < X then
      Writeln ('TOO LOW')
    else if G > X then
      Writeln ('TOO HIGH')
    else
      Writeln ('RIGHT');
    Inc(I);
  end;
end.


{1.2}
program One2T82;
{ -- This program will find #s that are the sum of 2 squares. }
  var
    I, J: Byte;
    A:    Array[1..50] of Boolean;

begin
  for I := 1 to 50 do A[I] := False;
  for I := 1 to 5 do
    for J := I to 7 do
      if I*I + J*J < 50 then A[I*I + J*J] := True;

  for I := 1 to 50 do
    if A[I] then Write (I, ',');
  Writeln;
end.
```

```
{1.3}
program One3T82;
{ -- This program will sum numbers divisible by 14. }
  var
    I: Integer;
    S: LongInt;

begin
  for I := 100 to 1000 do
    if I mod 14 = 0 then S := S + I;
  Writeln (S);
end.


{1.4}
program One4T82;
{ -- This program will add 2 random times. }
  var
    I, M, H:   Byte;
    Min, Hour: Array [1..2] of Byte;

begin
  Randomize;
  for I := 1 to 2 do begin
    Hour[I] := Random(12) + 1;
    Min[I]  := Random(60);
    Write (Hour[I], ':');
    if Min[I] < 10 then Write ('0');
    Writeln (Min[I]);
  end;
  Writeln ('-----');
  M := Min[1] + Min[2];  H := 0;
  if M > 59 then begin
    M := M - 60;  H := 1;
  end;
  H := H + Hour[1] + Hour[2];
  if H > 12 then H := H - 12;
  Write (H, ':');
  if M < 10 then Write ('0');
  Writeln (M);  Writeln;
end.
```

```
{1.5}
program One5T82;
{ -- This program will compute roots of equation. }
  var
    A, B, C, S: Integer;

begin
  Write ('Enter a, b, c: ');  Readln (A, B, C);
  S := B*B - 4*A*C;
  if S < 0 then
    Writeln ('COMPLEX')
  else begin
    Write ( (-B - Sqrt(S)) / (2 * A) : 4:2, '  ');
    Writeln ( (-B + Sqrt(S)) / (2 * A) : 4:2);
  end;
end.


{1.6}
program One6T82;
{ -- This program will print prime factors. }
  var
    N, I, J: Byte;
    Prime:   Boolean;

begin
  Write ('Enter number: ');  Readln (N);
  for I := 2 to N do
    if N mod I = 0 then begin
      J := 2;  Prime := True;
      while (J <= Trunc(Sqrt(I))) and Prime do begin
        if I mod J = 0 then Prime := False;
        Inc(J);
      end;
      if Prime then Write(I, '  ');
    end;
  Writeln;
end.


{1.7}
program One7T82;
{ -- This program will calculate future value of investment. }
  var
    P, i:    Real;
    J, N, Y: Integer;

begin
  Write ('Enter P, i, N, Y: ');  Readln (P, i, N, Y);
  for J := 1 to N * Y do
    P := P + P * i / N;
  Writeln ('$', Round(P * 100) / 100 :5:2);
end.
```

```
{1.8}
program One8T82;
{ -- Ths program will find 3 #s whose sum is 43. }
  var
    I, J, K: LongInt;

begin
  for I := 1 to 41 do
    for J := 1 to 42 - I do begin
      K := 43 - I - J;
      if I*I*I + J*J*J + K*K*K = 17299 then begin
        Writeln (I, '  ', J, '  ', K);  Exit;
      end;
    end;
end.


{1.9}
program One9T82;
{ -- This program will print a symbol for 45 seconds. }
uses Crt;
  var
    Ch: Char;

begin
  Write ('Enter a symbol: ');  Readln (Ch);
  ClrScr;  Write(Ch);
  Delay (45000);
  ClrScr;
end.



{1.10}
program One10T82;
{ -- This program will convert decimal to fraction. }
  var
    Dec:              String[12];
    L, N, D, I, Code: Integer;

begin
  Write ('Enter decimal: ');  Readln (Dec);
  L   := Length(Dec) - 1;
  Dec := Copy (Dec, 2, L);
  Val (Dec, N, Code);  D := 1;
  for I := 1 to L do   D := D * 10;
  for I := N downto 1 do
    if (N mod I = 0) and (D mod I = 0) then begin
      Writeln (N div I, '/', D div I);  Exit;
    end;
end.
```

```
{1.11}
program One11T82;
{ -- This program will move an asterisk by pressing keys. }
uses Crt;
  var
    R, C: Integer;
    Ch:   Char;

begin
  ClrScr;  R := 10;  C := 40;
  while Ch <> ' ' do begin
    GotoXY (C, R);  Write ('*');
    Ch := ReadKey;
    if Ch in ['U', 'D', 'L', 'R'] then begin
      GotoXY (C, R);  Write (' ');
      if Ch = 'U' then Dec(R);
      if Ch = 'D' then Inc(R);
      if Ch = 'L' then Dec(C);
      if Ch = 'R' then Inc(C);
    end;
  end;
end.
```

```
{2.1}
program Two1T82;
{ -- This program will print day of week of a date. }
  const
    M: Array [1..12] of Integer =
       (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  var
    I, Mo, Da, S, X: Integer;

begin
  Write ('Enter month, day: ');  Readln (Mo, Da);
  S := 0;
  for I := 1 to Mo - 1 do S := S + M[I];
  S := S + Da;
  X := S mod 7;
  Writeln ( Copy('THUFRISATSUNMONTUEWED', X*3 + 1, 3) );
end.


{2.2}
program Two2T82;
{ -- This program will calculate the area of a polygon. }
  var
    N, I, Sum: Integer;
    X, Y:      Array[1..9] of Integer;

begin
  Write ('Enter n: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter vertex (X, Y): ');  Readln (X[I], Y[I]);
  end;
  X[N+1] := X[1];  Y[N+1] := Y[1];  Sum := 0;
  for I := 1 to N do
    Sum := Sum + X[I] * Y[I+1] - Y[I] * X[I+1];
  Writeln ('AREA = ', Abs(Sum) / 2  :4:1);
end.
```

```
{2.3}
program Two3T82;
{ -- This program will find 5 digit number. }
{ -- Strategy:  # is less than 25000 because 4 * # would be
                  a 6 digit # otherwise.
               # can't be 1XXXY since 4 * Y can't give us
                  a 1 in the units place.
               # must therefore begin with 2 and end with
                  8 since 4*8 = 32.  So we can step 10.   }
  var
    I:     LongInt;
    J:     Integer;
    N, S:  String[5];
    Found: Boolean;

begin
  I := 20008;
  repeat
    Str (I, N);  Str (I*4, S);  Found := True;
    for J := 1 to 5 do
      if Copy(N, J, 1) <> Copy(S, 6-J, 1) then
        Found := False;
    if Found then
      Writeln (I)
    else
      I := I + 10;
  until (I >= 24998) or Found;
end.



{2.4}
program Two4T82;
{ -- This program will find interesting numbers. }
  var
    I, J, K, Num, Pow: Integer;

begin
  for I := 1 to 9 do
    for J := 0 to 9 do
      for K := 0 to 9 do begin
        Num := I * 100 + J * 10 + K;
        Pow := I*I*I + J*J*J + K*K*K;
        if (Num = Pow) and (Num <> 153) then
          Write (Num :5);
      end;
      Writeln;
end.
```

```
{2.5}
program Two5T82;
{ -- This program will make user's name zig zag. }
uses Crt;
  var
    I, X, L, S: Byte;
    Nam:        String[20];
    Ch:         Char;

begin
  Write ('Enter name: ');  Readln (Nam);  ClrScr;
  L := Length (Nam);
  X := Trunc(159 / (L-1));
  for I := 1 to L do begin
    Ch := Nam[I];
    S := (I - 1) * X;
    if S > 79 then S := 159 - S;
    Writeln (' ': S, Ch);
  end;
end.

{2.6}
program Two6T82;
{ -- This program will print a stick figure. }
uses Crt;
  var
    R, C, I, K: Byte;
    Inc:        Real;
    A:          Char;

begin
  R := 5;  C := 12;
  repeat
    for I := 0 to 5 do begin
      ClrScr;
      Writeln (' *     ***** ');
      Writeln ('  *    *   * ');
      Writeln ('   *   ***** ');
      Writeln ('    **   *   ');
      Writeln ('      *****  ');
      Writeln ('         *   ');
      Writeln ('         *   ');
      Writeln ('        * *  ');
      Writeln ('       *   * ');
      Writeln ('      *     *');
      Inc := (R - I) / 7;
      For K := 0 to 6 do begin
        GotoXY (C+K, R-Trunc(Inc*K));  Write ('*');
      end;
      Delay(100);
    end;
    A := ReadKey;
  until A = Char(27);
end.
```

```
{2.7}
program Two7T82;
{ -- This program will display permutations of letters. }
uses Crt;
  var
    N, I, X: Integer;
    A:       Array [1..8] of Char;
    Temp:    Char;

begin
  Randomize;
  Write ('How many letters: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter letter: ');  Readln (A[I]);
  end;
  repeat
    for I := 1 to N do begin
      X := Random(N) + 1;
      Temp := A[X];  A[X] := A[I];  A[I] := Temp;
    end;
    for I := 1 to N do Write (A[I]);
    Writeln; Delay(100);
  until Keypressed;
end.
```

```
{2.8}
program Two8T82;
{ -- This program will drill typying skills. }
uses Crt;
  var
    I, X, J: Integer;
    S:       LongInt;
    A, B:    Array[1..4] of Char;
    Ch:      Char;
    Wrong:   Boolean;

begin
  Randomize;
  for I := 1 to 4 do begin
    X := Random(58) + 33;
    A[I] := Chr(X);
    Write (A[I], ' ');
  end;
  Writeln;  J := 1;  S := 0;
  while J < 5 do begin
    repeat
      Inc(S);
    until Keypressed;
    Ch := ReadKey;  B[J] := Ch;
    Write (Ch, ' ');
    Inc(J);
  end;
  Writeln;  Writeln;  Wrong := False;
  for I := 1 to 4 do
    if A[I] <> B[I] then begin
      Writeln (A[I], ' --- ', B[I], '  NO');
      Wrong := True;
    end;
  if Not Wrong then Writeln (S div 30000, ' SECONDS');
end.
```

```
{2.9}
program Two9T82;
{ -- This program will return change in fewest coins. }
  const
    Nam: Array [1..8] of String[9] =
      ('$20', '$10', '$5', 'DOLLARS', 'QUARTERS', 'DIMES',
       'NICKELS', 'PENNIES');
    Amount: Array [1..8] of Integer =
      (2000, 1000, 500, 100, 25, 10, 5, 1);
  var
    P: Real;
    N, D, I, X: Integer;

begin
  Write ('Enter price $: ');  Readln (P);
  Write ('Enter denomination $: ');  Readln (D);
  N := D * 100 - Trunc(P * 100 + 0.1);
  for I := 1 to 8 do begin
    X := N div Amount[I];
    if X > 0 then Writeln (X, ' ', Nam[I]);
    N := N - X * Amount[I];
  end;
end.


{2.10}
program Two10T82;
{ -- This program will make unit conversions. }
  const
    A: Array[1..5] of String[2] =
      ('IN', 'FT', 'FT', 'YD', 'MI');
    B: Array[1..5] of String[2] =
      ('CM', 'CM', 'M ', 'M ', 'KM');
  var
    I, X: Byte;
    N, S: Real;

begin
  for I := 1 to 5 do
    Writeln (I, '  ', A[I], ' -> ', B[I]);
  Write ('Enter Choice #: ');  Readln (X);
  Write ('Enter ', A[X], ': ');  Readln (N);
  S := N * 2.54;
  if X = 1 then Write (S :6:2);
  if X = 2 then Write (S * 12 :6:2);
  if X = 3 then Write (S * 12 / 100 :6:2);
  if X = 4 then Write (S * 36 / 100 :6:2);
  if X = 5 then Write (S * 5280 * 12 / 100000.0 :6:2);
  Writeln (' ', B[X]);
end.
```

```
{2.11}
program Two11T82;
{ -- This program will find A^B x C^D = ABCD }
  var
    A, B, C, D, J, APow, CPow, Num: LongInt;

begin
  for A := 1 to 9 do
    for B := 0 to 9 do
      for C := 0 to 9 do
        for D := 0 to 9 do begin
          APow := 1;  CPow := 1;
          for J := 1 to B do APow := APow * A;
          for J := 1 to D do CPow := CPow * C;
          Num := A*1000 + B*100 + C*10 + D;
          if APow * CPow = Num then begin
            Writeln ('A=', A, '  B=', B, '  C=', C, '  D=',D);
            Exit;
          end;
        end;
end.



{2.12}
program Two12T82;
{ -- This program calculates days between 2 dates. }
  const
    Days: Array[1..12] of Integer =
      (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  var
    M1, D1, M2, D2, I, S: Integer;

begin
  Write ('Enter Month1, Day1: ');  Readln (M1, D1);
  Write ('Enter Month2, Day2: ');  Readln (M2, D2);
  S := 0;
  for I := M1 to M2-1 do S := S + Days[I];
  Writeln (S + D2 - D1, ' DAYS');
end.
```

```
{2.13}
program Two13T82;
{ -- This program will print a check. }
uses Crt;
  const
    Mo: Array [1..12] of String[5] =
      ('JAN.', 'FEB.', 'MAR.', 'APRIL', 'MAY', 'JUNE',
       'JULY', 'AUG.', 'SEPT.', 'OCT.', 'NOV.', 'DEC.');
    Words: Array[1..27] of String[10] =
      ('ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN',
       'EIGHT', 'NINE', 'TEN', 'ELEVEN', 'TWELVE', 'THIRTEEN',
       'FOURTEEN', 'FIFTEEN', 'SIXTEEN', 'SEVENTEEN',
       'EIGHTEEN', 'NINETEEN', 'TWENTY-', 'THIRTY-', 'FOURTY-',
       'FIFTY-', 'SIXTY-', 'SEVENTY-', 'EIGHTY-', 'NINETY-');
  var
    I, M, D, Y, S, T, X, Cent: Integer;
    Nam: String[20];
    N:   Real;

begin
  Write ('Enter month, day, year: ');  Readln (M, D, Y);
  Write ('Enter amount $:');  Readln (N);
  Write ('Enter payee: ');    Readln (Nam);
       { -- Display check border }
  ClrScr;
  for I := 1 to 60 do Write ('*');
  for I := 1 to 7 do begin
    GotoXY (1, I+1);  Write ('*');
    GotoXY (60,I+1);  Write ('*');
  end;
  Writeln;
  for I := 1 to 60 do Write ('*');
       { -- Display date, Name, and amount }
  GotoXY (45, 2);  Write (Mo[M], ' ', D, ', 19', Y);
  GotoXY (5, 4);   Write ('PAY TO THE');
  GotoXY (5, 5);   Write ('ORDER OF  ', Nam);
  GotoXY (50, 5);  Write ('$', N:5:2);
  GotoXY (3, 7);
       { -- Display amount in words }
  Cent := Trunc( (N - Int(N)) * 100 + 0.1);  S := 1000;  T := 0;
  for I := 2 downto 0 do begin
    S := S div 10;  X := Trunc(N/S + 0.001);
    if (I = 2) and (X > 0) then
      Write (Words[X], ' HUNDRED ');
    if (I = 1) and (X > 1) then
      Write (Words[18+X]);
    if (I = 1) and (X = 1) then T := 1 else T := 0;
    if I = 0 then
      Write (Words[T*10+X]);
    N := Int(N - X * S + 0.001);
  end;
  Write (' AND ', Cent, '/100 DOLLARS');
end.
```

```
{3.1}
program Thr1T82;
{ -- This program will play mastermind. }
{ -- The computer will randomly select four of the six colors.  }
{ -- The user must guess this combination of four colors.       }
{ -- BLACK indicates that a color is in the right position.     }
{ -- WHITE indicates a color is right but in the wrong position.}
uses Crt;
  const
    Co: Array [1..6] of String[2] =
      ('W', 'Y', 'R', 'G', 'BL', 'BK');
  var
    I, J, K, W, Bk, X: Integer;
    A, B, C:           Array[1..6] of String[2];

begin
  Randomize;
  for I := 1 to 4 do begin
    X := Random(6) + 1;  A[I] := Co[X];
  end;

  ClrScr;  Writeln ('GUESS: W, Y, R, G, BL, BK');
  for K := 1 to 10 do begin
    W := 0;  Bk := 0;
    for I := 1 to 4 do begin
      GotoXY (I*6, K*2);  Readln (B[I]);
    end;
    for I := 1 to 4 do C[I] := A[I];
    for I := 1 to 4 do
      if C[I] = B[I] then begin
        Inc(Bk);  B[I] := '';  C[I] := ' ';
      end;
    for I := 1 to 4 do
      for J := 1 to 4 do
        if C[I] = B[J] then begin
          Inc(W);  B[J] := '';  C[I] := ' ';
        end;
    { -- Black pegs = Correct color and correct position }
    { -- White pegs = Correct color but wrong position }
    GotoXY (40, K*2);
    Write ('BLACKS = ', Bk, '   WHITES = ', W);
    if Bk = 4 then begin
       Writeln; Writeln ('YOU WIN IN ', K, ' TURNS');  Exit;
    end;
  end;  { -- for K }
  Writeln;  Writeln ('YOU LOSE');
  for I := 1 to 4 do Write (A[I], ' ');
end.
```

```
{3.2}
program Thr2T82;
{ -- This program will plot points on a new axis. }
uses Crt;
  var
    X1, Y1, X2, Y2, IT, N, I, R, C: Integer;
    X, Y:                           Array[1..9] of Integer;

begin
  Write ('Enter end point of x-axis: ');  Readln (X1, Y1);
  Write ('Enter end point of y-axis: ');  Readln (X2, Y2);
  Write ('Enter increment: ');  Readln (IT);
  Write ('How many points: ');  Readln (N);
  for I := 1 to N do begin
    Write ('Enter point: ');  Readln (X[I], Y[I]);
  end;
  ClrScr;  R := 3;  C := 1;
  Writeln ('INTERSECTION AT (', X2, ',', Y1, ')');
  Writeln;
  I := Y1;
  repeat
    Write ('*');  I := I + IT;
  until I > Y2;
  I := X2 + IT; Writeln;
  repeat
    Writeln ('*');  I := I + IT;
  until I > X1;
  for I := 1 to N do begin
    GotoXY (C + (Y[I]-Y1) div IT, R + (X[I]-X2) div IT);
    Write ('+');
  end;
end.
```

```
{3.3}
program Thr3T82;
{ -- This program will generate magic squares. }
{ -- Correctly for odd matrices and for a 4 x 4. }
uses Crt;
  var
    N, X, Y, I, J, S: Integer;
    A: Array [1..12, 1..12] of Integer;

begin
  ClrScr;
  Write ('Enter size: ');  Readln (N);
  Writeln;  S := 0;
  if N mod 2 = 1 then begin  { -- routine for odd Matrix }
    for X := 1 to N do
      for Y := 1 to N do
        A[X,Y] := 0;
    X := 1;  Y := (N+1) div 2;  A[X,Y] := 1;
    for I := 2 to N*N do begin
      Dec(X);  Dec(Y);
      if X = 0 then X := N;
      if Y = 0 then Y := N;
      if A[X,Y] = 0 then
        A[X,Y] := I
      else begin
        X := X + 2;  Inc(Y);
        if X > N then X := X - N;
        if Y > N then Y := 1;
        A[X,Y] := I;
      end;
    end;  { -- for I }
    end   { -- begin }
  else    { -- Routine for Even Matrix (4x4) }
    for I := 1 to N do
      for J := 1 to N do begin
        S := S + 1;
        if (I = J) or (I = N+1-J) then
          A[I,J] := S
        else
          A[I,J] := N*N + 1 - S;
      end;

  for I := 1 to N do
    for J := 1 to N do begin
      GotoXY (J*4, I*2);  Write (A[I,J]);
    end;
  Writeln;  Writeln ('MAGIC NUMBER = ', (N*N*N + N) div 2);
end.
```

```
{3.4}
program Thr4T82;
{ -- This program will add and multiply 2 Roman Numerals. }
  const
    RN: Array[1..7] of Char =
      ('M', 'D', 'C', 'L', 'X', 'V', 'I');
    RV: Array[1..7] of Integer =
      (1000, 500, 100, 50, 10, 5, 1);
  var
    I, E, L, Ar, I1, I2, J, K, XX, Num: Integer;
    Rom, R:  Array [1..2] of String[15];
    A, N:    Array [1..2] of Integer;
    Ch, NCh: String[1];
    X:       Real;

begin
  for E := 1 to 2 do begin
    Write ('Enter Roman Numeral: ');  Readln (Rom[E]);
    L := Length(Rom[E]);  I := 1;  Ar := 0;
    while I < L do begin
      Ch := Copy (Rom[E], I, 1);
      I1 := 1;  while Ch <> RN[I1] do Inc(I1);
      NCh:= Copy (Rom[E], I+1, 1);
      I2 := 1;  while NCh <> RN[I2] do Inc(I2);
      if I1 <= I2 then
        Ar := Ar + RV[I1]
      else begin
        Ar := Ar + RV[I2] - RV[I1];  Inc(I);  end;
      Inc(I);
    end;
    if I <= L then begin  { -- Last numeral not done }
      Ch := Copy (Rom[E], I, 1);
      I1 := 1;  while Ch <> RN[I1] do Inc(I1);
      Ar := Ar + RV[I1];
    end;
    A[E] := Ar;
  end;  { -- for E }
  { -- Convert Arabic numbers to Roman Numerals }
  N[1] := A[1] + A[2];  N[2] := A[1] * A[2];
  R[1] := '';  R[2] := '';
  for K := 1 to 2 do begin
    Num := N[K];
    for I := 1 to 7 do begin
      X := Num / RV[I];
      if (X<2) and (X>=9/5) and (I in [2,4,6]) then { -- next }
      else begin
        XX := Trunc(X);
        if XX = 9 then R[K] := R[K] + RN[I] + RN[I-2]
        else
        if XX = 4 then R[K] := R[K] + RN[I] + RN[I-1]
        else
        if XX > 0 then
          for J := 1 to XX do
            R[K] := R[K] + RN[I];
        Num := Num - RV[I] * XX;
```

```
      end;
    end;  { -- for I }
  end;  { -- for K }
  { -- Display sum and product }
  Writeln (Rom[1], ' + ', Rom[2], ' = ', R[1]);
  Writeln (A[1], ' + ', A[2], ' = ', N[1]);
  Writeln (Rom[1], ' * ', Rom[2], ' = ', R[2]);
  Writeln (A[1], ' * ', A[2], ' = ', N[2]);
end.


{3.5}
program Thr5T82;
{ -- This program will find 4 digit squumbers. }
  var
    I, L, R, X, Code: Integer;
    Ist:              String[4];

begin
  for I := 1000 to 9999 do begin
    Str (I, Ist);
    Val (Copy(Ist, 1, 2), L, Code);
    Val (Copy(Ist, 3, 2), R, Code);
    X := L + R;
    if X * X = I then Writeln (I);
  end;
end.

{3.6}
program Thr6T82;
{ -- This program should play Nim with a user.           }
{ -- HOWEVER, since the rules are not given with this    }
{ -- problem, it is very difficult to write the program. }
begin
end.


{3.7}
program Thr7T82;
{ -- This program will determine where a # falls in a list. }
  var
    A:      Array [1..16] of Integer;
    I, Num: Integer;

begin
  for I := 1 to 16 do begin
    Write ('Enter #: ');  Readln (A[I]);
  end;
  Write ('Enter another number: ');  Readln (Num);
  I := 1;
  while A[I] <> Num do Inc(I);
  Writeln ('BETWEEN ', A[I-1], ' AND ', A[I+1]);
end.
```

```
{3.8}
program Thr8T82;
{ -- This BONUS program will guess the user's state. }
  const
    State: Array[1..50] of String[14] =
    ('ALABAMA','ALASKA','ARIZONA','ARKANSAS','CALIFORNIA',
    'COLORADO','CONNECTICUT','DELEWARE','FLORIDA','GEORGIA',
    'HAWAII','IDAHO','ILLINIOS','INDIANA','IOWA','KANSAS',
    'KENTUCKY','LOUISIANA','MAINE','MARYLAND','MASSACHUSETTS',
    'MICHIGAN','MINNESOTA','MISSISSIPPI','MISSOURI','MONTANA',
    'NEBRASKA','NEVADA','NEW HAMPSHIRE','NEW JERSEY','NEW YORK',
    'NEW MEXICO','NORTH CAROLINA','NORTH DAKOTA','OHIO',
    'OKLAHOMA','OREGON','SOUTH CAROLINA','SOUTH DAKOTA',
    'PENNSYLVANIA','RHODE ISLAND','TENNESSEE','TEXAS','UTAH',
    'VERMONT','VIRGINIA','WASHINGTON','WEST VIRGINIA',
    'WISCONSIN','WYOMING');
  var
    I, G, B, M, E: Integer;
    A:             String[3];

begin
  G := 1;  B := 1;  M := 25;  E := 50;
  repeat
    Write (G, '- IS YOUR STATE ALPHABETICALLY BEFORE ', State[M]);
    Writeln;
    Write ('Enter YES or NO: ');  Readln (A);
    if (A = 'YES') and (B+1 = M) then begin
      Writeln (State[B], ' IS IT');  Exit;  end;
    if (A = 'NO') and (M = E) then begin
      Writeln (State[M], ' IS IT');  Exit;  end;
    if A = 'YES' then begin
      E := M - 1;  M := M - Round((M - B) / 2);  end
    else begin
      B := M;      M := M + Round((E - M) / 2);  end;
    Inc(G);
  until G > 12;
end.
```