

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '91 }
{ -- PASCAL PROGRAM SOLUTIONS }

{1.1}
program One1T91;
{ -- This program will display a phrase as a rectangle. }
uses Crt;
  const
    A = 'COMPUTER CONTEST 1991';
  var
    I, L: Byte;

begin
  ClrScr;
  Writeln(A);
  L := Length(A);
  for I := 2 to L - 1 do begin
    GotoXY(1, I); Write (Copy(A, I, 1));
    GotoXY(L, I); Write (Copy(A, L-I+1, 1));
  end;
  Writeln;
  for I := L downto 1 do
    Write (Copy(A, I, 1));
end.

{1.2}
program One2T91;
{ -- This program will display 2 random #s and their sum. }
  var
    X, Y: Integer;

begin
  Randomize;
  X := Random(19) - 9;
  Y := Random(19) - 9;
  Writeln (X, ' + ', Y, ' = ', X + Y);
end.
```

```
{1.3}
program One3T91;
{ -- This program prints the total point score for a team. }
var
  I, P, Sum: Byte;
  Nam:      String[20];
```

```
begin
  Sum := 0;
  Write ('Enter name: '); Readln (Nam);
  for I := 1 to 3 do begin
    Write ('Enter # of ', I, ' point programs: ');
    Readln (P);
    Sum := Sum + P * I;
  end;
  Writeln (Nam, ' SCORED ', Sum, ' POINTS');
end.
```

```
{1.4}
program One4T91;
{ -- This program displays a spreadsheet. }
uses Crt;
var
  I: Byte;
```

```
begin
  ClrScr;
  Writeln ('  A B C D E F G H I J K L M N O P Q R S T');
  for I := 1 to 20 do
    Writeln (I:2);
end.
```

```
{1.5}
program One5T91;
{ -- This program determines the number of teams competing. }
var
  X: Integer;
```

```
begin
  Write ('Enter number of students: '); Readln (X);
  Writeln (X div 4, ' TEAMS');
end.
```

```
{1.6}
program One6T91;
{ -- This program displays a word twice intersecting at a letter.}
uses Crt;
var
  A:   String[12];
  L:   String[1];
  X, I: Byte;

begin
  Write ('Enter word: '); Readln (A);
  Write ('Enter letter: '); Readln (L);
  X := Pos(L, A);
  ClrScr;
  GotoXY (1, X); Writeln (A);
  for I := 1 to Length(A) do begin
    GotoXY (X, I); Write (Copy(A, I, 1));
  end;
end.
```

```
{1.7}
program One7T91;
{ -- This program displays fields from an account key. }
var
  A: String[20];

begin
  Write ('Enter account key: '); Readln (A);
  Writeln ('ORGANIZATION ', Copy(A, 1, 3));
  Writeln ('BRANCH ', Copy(A, 4, 3));
  Writeln ('DEALER ', Copy(A, 7, 4));
  Writeln ('CLASS ', Copy(A, 11, 3));
  Writeln ('UNIT ', Copy(A, 14, 6));
end.
```

```
{1.8}
program One8T91;
{ -- This program displays the # of job steps in JCL. }
var
  L: String[5];
  S: Byte;

begin
  Write ('Enter line: '); Readln (L); S := 0;
  while L <> '//' do begin
    if L = 'EXEC' then Inc(S);
    Write ('Enter line: '); Readln (L);
  end;
  Writeln (S, ' JOB STEPS');
end.
```

```
{1.9}
program One9T91;
{ -- This program will replace MAN with PERSON. }
var
  S: String[100];
  M: String[3];
  I: Byte;

begin
  Write ('Enter sentence: '); Readln (S);
  for I := 1 to Length(S) do begin
    M := Copy(S, I, 3);
    if M = 'MAN' then begin
      Write ('PERSON'); I := I + 2; end
    else if M = 'MEN' then begin
      Write ('PERSONS'); I := I + 2; end
    else
      Write (Copy(S, I, 1));
  end;
end.
```

```
{1.10}
program One10T91;
{ -- This program determines the winner of two computer teams. }
var
  N1, N2: String[20];
  T1, T2, TI1, TI2: Integer;
  P1, P2, Pen1, Pen2, H1, H2, M1, M2: Byte;

begin
  Write ('Enter team name: '); Readln (N1);
  Write ('Enter points, time, penalties: ');
  Readln (P1, T1, Pen1);
  Write ('Enter team name: '); Readln (N2);
  Write ('Enter points, time, penalties: ');
  Readln (P2, T2, Pen2);

  if P1 > P2 then
    Write (N1)
  else if P2 > P1 then
    Write (N2)
  else begin
    H1 := T1 div 100; M1 := T1 mod 100;
    H2 := T2 div 100; M2 := T2 mod 100;
    TI1 := H1 * 60 + M1 + Pen1 * 5;
    TI2 := H2 * 60 + M2 + Pen2 * 5;
    if TI1 < TI2 then
      Write (N1)
    else
      Write (N2);
  end;
  Writeln (' WINS');
end.
```

```
{2.1}
program Two1T91;
{ -- This program displays a pyramid of consecutive numbers. }
var
  N, S, I, J: Byte;

begin
  Write ('Enter N: '); Readln (N);
  S := 1; I := 0;
  while S < N do begin
    Inc(I);
    Write (' ': 20 - I * 2);
    for J := 1 to I do begin
      if S < 10 then Write ('0');
      Write (S, ' ');
      Inc(S);
    end;
    Writeln;
  end;
end.
```

```
{2.2}
program Two2T91;
{ -- This program will line up numbers with decimal points. }
var
  I, X, Code: Integer;
  A:          Array [1..5] of String[9];
  Y, Sum:     Real;

begin
  for I := 1 to 5 do begin
    Write ('Enter #: '); Readln (A[I]);
  end;
  Sum := 0;
  for I := 1 to 5 do begin
    X := Pos('.', A[I]);
    Writeln (' ': 6 - X, A[I]);
    Val(A[I], Y, Code);
    Sum := Sum + Y;
  end;
  Writeln (' -----');
  Writeln (Sum: 10:4);
end.
```

```
{2.3}
program Two3T91;
{ -- This program will convert BASIC to COBOL. }
var
  S: String[80];
  M: String[1];
  MN: String[2];
  I: Byte;

begin
  Write ('Enter statement: '); Readln (S);
  for I := 1 to Length(S) do begin
    M := Copy(S, I, 1);
    MN := Copy(S, I, 2);
    if (MN = '<=') or (MN = '=<') then
      begin
        Write ('IS NOT GREATER THAN');
        Inc(I);
      end
    else if (MN = '>=') or (MN = '=>') then
      begin
        Write ('IS NOT LESS THAN');
        Inc(I);
      end
    else if (MN = '<>') or (MN = '><') then
      begin
        Write ('IS NOT EQUAL TO');
        Inc(I);
      end
    else if (M = '>') then
      Write ('IS GREATER THAN')
    else if (M = '<') then
      Write ('IS LESS THAN')
    else if (M = '=') then
      Write ('IS EQUAL TO')
    else
      Write (M);
  end;
end.
```

```

{2.4}
program Two4T91;
{ -- This program ranks teams in a league. }
var
  N, I, J, R, X: Integer;
  Na: Array [1..9] of String[20];
  W, L: Array [1..9] of Integer;
  T: String[20];

begin
  Write ('Enter N: '); Readln (N);
  for I := 1 to N do begin
    Write ('Enter team: '); Readln(Na[I]);
    Write ('Enter wins, losses: '); Readln (W[I], L[I]);
  end;
  for I := 1 to N - 1 do
    for J := I + 1 to N do
      if (W[I] < W[J]) or ((W[I] = W[J]) and (Na[I] > Na[J])) then
        begin
          X := W[I]; W[I] := W[J]; W[J] := X;
          X := L[I]; L[I] := L[J]; L[J] := X;
          T := Na[I]; Na[I] := Na[J]; Na[J] := T;
        end;
  end;
  for I := 1 to N do begin
    if W[I] = W[I - 1] then
      Write (R)
    else begin
      Writeln; Write(I); R := I;
    end;
    Write (' ', Na[I], ' ': 14 - Length(Na[I]), W[I]);
    Writeln (' ', ' ', L[I]);
  end;
end.

```

```

{2.5}
program Two5T91;
{ -- This program will guess a secret number within 7 tries. }
var
  Increment, Guess, G: Byte;
  A: Char;

begin
  Increment := 64; Guess := 64; G := 0; A := ' ';
  while A <> 'R' do begin
    Inc(G);
    Writeln ('GUESS ', G, ': ', Guess);
    Write ('Enter H, L, or R: '); Readln (A);
    Increment := Increment div 2;
    if A = 'L' then
      Dec(Guess, Increment);
    if A = 'H' then
      Inc(Guess, Increment);
  end;
end.

```

```
{2.6}
program Two6T91;
{ -- This program prints text in pyramid form. }
var
  A, Lin:   String[255];
  I, L, PL: Byte;
  MD:      String[1];

begin
  Write ('Enter text: '); Readln (A);
  L := Length(A);  I := 1;  PL := 0;  Lin := '';
  while I <= L do begin
    MD := Copy(A, I, 1);
    if MD <> ' ' then
      Lin := Lin + MD
    else if Length(Lin) < PL + 2 then
      Lin := Lin + MD
    else begin
      PL := Length(Lin);
      Writeln (' ': 20 - (PL div 2), Lin);
      Lin := '';
    end;
    Inc(I);
  end;
  PL := Length(Lin);
  Writeln (' ': 20 - (PL div 2), Lin);
end.
```

```
{2.7}
program Two7T91;
{ -- This program displays a rectangle of asterisks. }
uses Crt;
var
  L, W, I, Row, Col: Byte;

begin
  Write ('Enter length, width: '); Readln (L, W);
  ClrScr;
  Col := (80 - L) div 2;  Row := (24 - W) div 2;
  GotoXY (Col, Row);
  for I := 1 to L do Write ('*');
  for I := 1 to W - 2 do begin
    GotoXY (Col, Row + I);  Write ('*');
    GotoXY (Col + L - 1, Row + I);  Write ('*');
  end;
  GotoXY (Col, Row + W - 1);
  for I := 1 to L do Write ('*');
end.
```



```

{2.8}
program Two8T91;
{ -- This program displays a bar graph for lengths. }
uses Crt;
var
  A:   Array [0..11] of Integer;
  I, J: Byte;
  Max: Integer;
  Inc: Real;
  T:   String[40];

begin
  Write ('Enter title: '); Readln (T); Max := 0;
  for I := 0 to 11 do begin
    Write ('Enter # for ', 1980 + I, ': '); Readln (A[I]);
    if A[I] > Max then Max := A[I];
  end;
  Inc := Max / 20.0;
  ClrScr;
  Writeln (' ': 3, T, ' ': 3, 'ASTERISK = ', Inc: 7:2);
  for I := 20 downto 1 do Writeln (I: 2);
  for I := 1 to 12 * 3 + 2 do Write ('-');
  Writeln; Write (' ': 2);
  for I := 0 to 11 do Write (80 + I: 3);
  for I := 0 to 11 do
    for J := 1 to Trunc(A[I] / Inc) do begin
      GotoXY (I * 3 + 5, 22 - J); Write ('*');
    end;
  GotoXY(1, 22);
end.

```

```

{2.9}
program Two9T91;
{ -- This program displays a store maintenance list. }
var
  I, I1, I2, F1, F2: Byte;
  AN, CN, DN:       Byte;
  A, C, D:          Array [1..9] of String[10];
  ID1, ID2:         Array [1..9] of String[4];
  Item1, Item2:     Array [1..9] of Char;

begin
  Write ('Enter # of entries in yesterday's file: ');
  Readln (F1);
  for I := 1 to F1 do begin
    Write ('Enter ID: '); Readln (ID1[I]);
    Write ('Enter item: '); Readln (Item1[I]);
  end;
  Write ('Enter # of entries in today's file: ');
  Readln (F2);
  for I := 1 to F2 do begin
    Write ('Enter ID: '); Readln (ID2[I]);
    Write ('Enter item: '); Readln (Item2[I]);
  end;
end.

```

```
end;
ID2[F2 + 1] := 'ZZZZ'; ID1[F1 + 1] := '    ';
I1 := 1; I2 := 1; AN := 0; CN := 0; DN := 0;
while (I1 <= F1) or (I2 <= F2) do
  if ID1[I1] = ID2[I2] then
    if Item1[I1] <> Item2[I2] then { -- Changed }
      begin
        Inc(CN);
        C[CN] := ID1[I1] + ' ' + Item1[I1] + ' ' + Item2[I2];
        Inc(I1); Inc(I2);
      end
    else { -- No change }
      begin
        Inc(I1); Inc(I2);
      end
    else
      if (ID1[I1] < ID2[I2]) and (I1 <= F1) then { -- Deleted }
        begin
          Inc(DN);
          D[DN] := ID1[I1] + ' ' + Item1[I1];
          Inc(I1);
        end
      else
        begin { -- Added }
          Inc(AN);
          A[AN] := ID2[I2] + ' ' + Item2[I2];
          Inc(I2);
        end;
      end;

Writeln; Writeln ('ADDED');
for I := 1 to AN do Writeln (A[I]);
Writeln; Writeln ('CHANGED');
for I := 1 to CN do Writeln (C[I]);
Writeln; Writeln ('DELETED');
for I := 1 to DN do Writeln (D[I]);
Writeln;
Writeln ('TOTAL ADDED = ', AN);
Writeln ('TOTAL CHANGED = ', CN);
Writeln ('TOTAL DELETED = ', DN);
end.
```

```
{2.10}
program Two10T91;
{ -- This program displays the contents of contest diskettes. }
uses Crt;
const
  Z: Array [1..6] of String[3] =
    ('PRB', 'JDG', 'PG1', 'PG2', 'BAS', 'PAS');
  X: Array [1..3] of String[3] = ('ONE', 'TWO', 'THR');
var
  I, J, K, P, Y, Tot: Byte;
  Year: String[4];
  YY: String[2];
  Ch: Char;

begin
  Write ('Enter year: '); Readln (Year);
  YY := Copy(Year, 3, 2);
  for I := 1 to 4 do
    for J := 1 to 3 do
      Writeln ('FHS', YY, '-', J, '.', Z[I]);

  Tot := 12;
  for I := 5 to 6 do
    for J := 1 to 3 do begin
      P := 10;
      if (YY = '80') and (J = 3) then P := 12;
      if (YY = '81') then P := 5;
      if (YY = '82') and (J = 2) then P := 12;
      if (YY = '82') and (J = 3) then P := 8;
      for K := 1 to P do begin
        Writeln (X[J], K, 'T', YY, '.', Z[I]);
        Inc(Tot);
        if Tot = 20 then begin
          Ch := ReadKey;
          Tot := 0;
        end;
      end;
    end;
  end; { -- for J }

end.
```

```

{3.1}
program Thr1T91;
{ -- This program simulates a baseball game. }
uses Crt;
var
  I, Inn, T, S, B, W, R, O, Wtot, Otot: Byte;
  Stot, Btot: Integer;
  Run:      Array [1..2] of Byte;

begin
  Randomize; ClrScr; Writeln; Write (' ': 7);
  for I := 1 to 9 do Write (I:3);
  Writeln (' SCORE');
  Write (' ': 8);
  for I := 1 to 34 do Write ('-');
  Writeln;
  Writeln ('TEAM A !', ' ': 27, '!!');
  Writeln ('TEAM B !', ' ': 27, '!!');
  Stot := 0; Btot := 0; Otot := 0; Wtot := 0;
  Run[1] := 0; Run[2] := 0;

  for Inn := 1 to 9 do
    for T := 1 to 2 do begin
      S := 0; B := 0; W := 0; R := 0; O := 0;
      while O < 3 do begin
        if Random < 0.4 then begin
          Inc(S); Inc(Stot); end
        else begin
          Inc(B); Inc(Btot);
        end;
        if S = 3 then begin
          Inc(O); Inc(Otot); S := 0; W := 0;
        end;
        if B = 4 then begin
          Inc(W); Inc(Wtot); B := 0; S := 0;
        end;
        if W = 4 then begin
          Inc(R); Inc(Run[T]); W := 3;
        end;
      end;
      GotoXY (6 + Inn * 3, 3 + T); Write (R:2);
    end; { -- for T }

  GotoXY (38, 4); Writeln (Run[1]: 3);
  GotoXY (38, 5); Writeln (Run[2]: 3);
  Writeln;
  Writeln ('TOTAL # OF STRIKES: ', Stot);
  Writeln ('TOTAL # OF BALLS: ', Btot);
  Writeln ('TOTAL # OF WALKS: ', Wtot);
  Writeln ('TOTAL # OF STRIKE OUTS: ', Otot);
end.

```

```
{3.2}
program Thr2T91;
{ -- This program displays the units digit in a power expression.}
var
  A, X: Array [1..3] of Integer;
  I, J, Pow, Sum, C: Integer;

begin
  Write ('Enter A, X: '); Readln (A[1], X[1]);
  Write ('Enter B, Y: '); Readln (A[2], X[2]);
  Write ('Enter C, Z: '); Readln (A[3], X[3]);
  Sum := 0;
  for I := 1 to 3 do begin
    Pow := 1;
    for J := 1 to X[I] do begin
      Pow := Pow * A[I];
      C := Pow div 10;
      Pow := Pow - C * 10;
    end;
    Sum := Sum + Pow;
  end;
  C := Sum div 10;
  Writeln (Sum - C * 10);
end.
```

```
{3.3}
program Thr3T91;
{ -- This program displays all digits in X ^ Y. }
var
  A: Array [1..200] of Integer;
  X, Y, I, J, Dig, C, CC: Integer;

begin
  Write ('Enter X, Y: '); Readln (X, Y);
  Dig := 1; A[1] := 1; C := 0;
  for I := 1 to Y do begin
    for J := 1 to Dig do begin
      A[J] := A[J] * X + C;
      C := A[J] div 10;
      A[J] := A[J] - C * 10;
    end;
    while C > 0 do begin
      CC := C div 10;
      Dig := Dig + 1;
      A[Dig] := C - CC * 10;
      C := CC;
    end;
  end;
  for I := Dig downto 1 do Write (A[I]);
end.
```

```

{3.4}
program Thr4T91;
{ -- This program assigns user LOGON IDs to names. }
var
  N, Fn, Mn, Ln, Init, In2, N2: Array [1..9] of String[20];
  T, I, J, M, F, Y, A, B:      Byte;
  C:                            Array [1..9] of Byte;
  MD:                           String[1];
  W, X:                          String[20];

begin
  Write ('Enter name: '); Readln (N[1]); T := 1;
  while N[T] <> 'END' do begin
    Inc(T);
    Write ('Enter name: '); Readln (N[T]);
  end;
  { -- Extract parts of name for initials }
  Dec(T);
  for I := 1 to T do begin
    W := ''; M := 0; F := 0;
    for J := 1 to Length(N[I]) do begin
      MD := Copy (N[I], J, 1);
      if MD <> ' ' then
        W := W + MD
      else
        if F = 1 then begin
          Mn[I] := W; M := 1; W := ''; end
        else begin
          Fn[I] := W; F := 1; W := ''; end;
        end; { -- for J }
      if M = 0 then Mn[I] := 'X';
      Ln[I] := W;
      Init[I] := Copy(Fn[I],1,1) + Copy(Mn[I],1,1) + Copy(Ln[I],1,1);
      In2[I] := Init[I]; N2[I] := Ln[I] + ' ' + Fn[I]; C[I] := I;
    end; { -- for I }
    { -- Sort Initials }
    for I := 1 to T - 1 do
      for J := I + 1 to T do
        if In2[I] > In2[J] then begin
          X := In2[I]; In2[I] := In2[J]; In2[J] := X;
          X := N2[I]; N2[I] := N2[J]; N2[J] := X;
          Y := C[I]; C[I] := C[J]; C[J] := Y;
        end;
    { -- Sort names within same initials and assign numbers. }
    J := 0;
    while J < T - 1 do begin
      I := J + 1; J := I + 1;
      while (In2[I] <> In2[J]) and (I < T) do begin
        Inc(I); Inc(J);
      end;
      while (In2[I] = In2[J]) do Inc(J);
      Dec(J);
      for A := I to J - 1 do
        for B := A + 1 to J do
          if N2[A] > N2[B] then begin

```

```
        X := N2[A];  N2[A] := N2[B];  N2[B] := X;
        Y := C[A];  C[A]  := C[B];  C[B]  := Y;
    end;
    { -- Assign numbers for middle initial }
    for A := I to J do
        Init[C[A]] := Copy(Init[C[A]],1,1) + Chr(48 + (A - I + 1))
                    + Copy(Init[C[A]],3,1);
    end; { -- while }
    for I := 1 to T do
        Writeln (N[I], ' ': 19 - Length(N[I]), 'SD', Init[I], '1');
    end.
```

```

{3.5}
program Thr5T91;
{ -- This program displays the digits 0 - 9 in enlarged form. }
{ 1   The data contains the }
{ 2 3 line segment #s (on the left) }
{ 4   that need to be displayed to }
{ 5 6 produce the corresponding }
{ 7   digits: 0,1,2,3,4,5,6,7,8,9. }
uses Crt;
const
  A: Array [0..9] of String[7] =
    ('123567', '36', '13457', '13467', '2346',
     '12467', '124567', '136', '1234567', '12346');
var
  N, I, J, X: Byte;

begin
  for N := 0 to 9 do begin
    ClrScr;
    for J := 1 to Length(A[N]) do begin
      X := Ord(A[N,J]) - Ord('0');
      Case X of
        1: begin
            GotoXY (1,1); for I := 1 to 11 do Write ('*');
          end;
        2: for I := 1 to 8 do begin
            GotoXY (1, I); Write ('*');
          end;
        3: for I := 1 to 8 do begin
            GotoXY (11, I); Write ('*');
          end;
        4: begin
            GotoXY (1,8); for I := 1 to 11 do Write ('*');
          end;
        5: for I := 1 to 8 do begin
            GotoXY (1, I+7); Write ('*');
          end;
        6: for I := 1 to 8 do begin
            GotoXY (11, I+7); Write ('*');
          end;
        7: begin
            GotoXY (1, 15); for I := 1 to 11 do Write ('*');
          end;
      end; { -- case }
    end; { -- next J }
    Delay (1000);
  end; { -- next N }
end.

```



```
{3.6}
program Thr6T91;
{ -- This program will evaluate an expression with (). }
var
  I, J, N, S, P: Byte;
  A:      String[50];
  Ch:     Char;
  P1, Num: Array [1..10] of Integer;
  SY:     Array [1..9]  of String[1];

begin
  Write ('Enter expression: '); Readln (A);
  P := 0; S := 0; N := 0;
  for I := 1 to Length(A) do begin
    Ch := A[I];
    if Ch = '(' then begin
      Inc(P); P1[P] := S + 1; end
    else if (Ch = '+') or (Ch = '-') then begin
      Inc(S); SY[S] := Ch; end
    else if (Ch >= '0') and (Ch <= '9') then begin
      Inc(N); Num[N] := Ord(Ch) - 48; end
    else if Ch = ')' then begin
      for J := P1[P] to S do begin
        if SY[J] = '-' then Num[J+1] := Num[J] - Num[J+1];
        if SY[J] = '+' then Num[J+1] := Num[J] + Num[J+1];
      end;
      N := P1[P]; Num[N] := Num[S + 1];
      S := P1[P] - 1; Dec(P);
    end;
  end;
  for I := 1 to S do begin
    if SY[I] = '-' then Num[I+1] := Num[I] - Num[I+1];
    if SY[I] = '+' then Num[I+1] := Num[I] + Num[I+1];
  end;
  Writeln (Num[N]);
end.
```

```

{3.7}
program Thr7T91;
{ -- This program displays the two pay days for a given month. }
const
  Mname: Array [1..12] of String[9] = ('JANUARY', 'FEBRUARY',
    'MARCH', 'APRIL', 'MAY', 'JUNE', 'JULY', 'AUGUST',
    'SEPTEMBER', 'OCTOBER', 'NOVEMBER', 'DECEMBER');
  Mon: Array [1..12] of Byte =
    (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
  Dname: Array [1..7] of String[9] = ('MONDAY', 'TUESDAY',
    'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY');
var
  I, T, H, Hol, Wkend, X, MNum: Byte;
  Mhol, Dhol: Array [1..12] of Byte;
  Day, Days: Array [1..2] of Integer;

begin
  H := 1;
  Write ('Enter holiday MM, DD: '); Readln (Mhol[H], Dhol[H]);
  while Mhol[H] > 0 do begin
    Inc(H);
    Write ('Enter holiday MM, DD: '); Readln (Mhol[H], Dhol[H]);
  end;
  Dec(H); Writeln;
  Write ('Enter month #: '); Readln (MNum); Writeln;
  while MNum > 0 do begin
    Days[1] := 0;
    for I := 1 to MNum - 1 do
      Days[1] := Days[1] + Mon[I];
    Day[1] := 15; Day[2] := Mon[MNum];
    Days[2] := Days[1] + Day[2];
    Days[1] := Days[1] + Day[1];
    for T := 1 to 2 do begin
      Hol := 1; Wkend := 1;
      { -- Decrement days counter if holiday or weekend. }
      while (Hol = 1) or (Wkend = 1) do begin
        Hol := 0; Wkend := 0;
        for I := 1 to H do
          if (Mhol[I] = MNum) and (Dhol[I] = Day[T]) then begin
            Dec(Day[T]);
            Dec(Days[T]); Hol := 1;
          end;
        X := Days[T] mod 7;
        if (X = 5) or (X = 6) then begin { -- Sat. or Sun. }
          Dec(Day[T]);
          Dec(Days[T]); Wkend := 1;
        end;
      end; { -- while }
      Writeln (Dname[X+1], ' ', Mname[MNum], ' ', Day[T]);
    end; { -- for T }
    Writeln; Write ('Enter month #: '); Readln (Mnum); Writeln;
  end; { -- while }
end.

```

```

{3.8}
program Thr8T91;
{ -- This program will display 3 x 3 magic squares. }
var
  Dig, Row, Col, I, J, P, Rot, X: Byte;
  A: Array [1..3,1..3] of Byte;

begin
  A[1,1] := 6; A[1,2] := 7; A[1,3] := 2;
  A[2,1] := 1; A[2,2] := 5; A[2,3] := 9;
  A[3,1] := 8; A[3,2] := 3; A[3,3] := 4;
  Write ('Enter digit: '); Readln (Dig);
  Write ('Enter row, col: '); Readln (Row, Col);
  Rot := 1;
  while (A[Row,Col] <> Dig) and (Rot < 4) do begin
    { -- Rotate outer numbers clockwise, at most 3 times }
    X := A[1,1]; A[1,1] := A[3,1]; A[3,1] := A[3,3];
    A[3,3] := A[1,3]; A[1,3] := X;
    X := A[1,2]; A[1,2] := A[2,1]; A[2,1] := A[3,2];
    A[3,2] := A[2,3]; A[2,3] := X;
    Inc(Rot);
  end;
  if A[Row,Col] <> Dig then begin
    Writeln ('NO SOLUTION'); Exit;
  end;
  for P := 1 to 2 do begin
    for I := 1 to 3 do begin
      for J := 1 to 3 do
        Write (A[I,J], ' ');
      Writeln;
    end;
    Writeln;
    if P = 1 then begin
      if (Row = 1) and (Col = 3) or (Row = 3) and (Col = 1) then
        begin
          X := A[2,1]; A[2,1] := A[3,2]; A[3,2] := X;
          X := A[1,1]; A[1,1] := A[3,3]; A[3,3] := X;
          X := A[1,2]; A[1,2] := A[2,3]; A[2,3] := X;
        end;
      if (Row = 1) and (Col = 1) or (Row = 3) and (Col = 3) then
        begin
          X := A[1,2]; A[1,2] := A[2,1]; A[2,1] := X;
          X := A[1,3]; A[1,3] := A[3,1]; A[3,1] := X;
          X := A[3,2]; A[3,2] := A[2,3]; A[2,3] := X;
        end;
      if (Row = 1) and (Col = 2) or (Row = 3) and (Col = 2) then
        begin
          X := A[1,1]; A[1,1] := A[1,3]; A[1,3] := X;
          X := A[2,1]; A[2,1] := A[2,3]; A[2,3] := X;
          X := A[3,1]; A[3,1] := A[3,3]; A[3,3] := X;
        end;
      if (Row = 2) and (Col = 1) or (Row = 2) and (Col = 3) then
        begin
          X := A[1,1]; A[1,1] := A[3,1]; A[3,1] := X;
          X := A[1,2]; A[1,2] := A[3,2]; A[3,2] := X;
        end;
    end;
  end;
end;

```

```
        X := A[1,3];  A[1,3] := A[3,3];  A[3,3] := X;
    end;
end;
end; { -- for P }
end.
```

```

{3.9}
program Thr9T91;
{ -- This program will display a pie graph. }
uses Crt;
const
  L: Array [1..3] of Char = ('A', 'D', 'N');
  PI: Real = 3.1415926;
var
  A: Array[1..21, 1..21] of Byte;
  P: Array[1..3] of Byte;
  I: Real;
  Ch: Char;
  J, K, R, X, Y, S, Sum, LSum: Integer;

begin
  Write ('Enter 3 percentages: '); Readln (P[1], P[2], P[3]);
  ClrScr;
  for J := 1 to 21 do
    for K := 1 to 21 do
      A[J, K] := 0;
    { -- Draw Circle }
    I := -PI / 2.0;
    while I < 3 / 2 * PI do begin
      X := Trunc(Cos(I) * 10); Y := Trunc(Sin(I) * 10);
      GotoXY (11 + X, 11 + Y); Write ('*');
      A[11 + X, 11 + Y] := 1; I := I + 0.1;
    end;
    { -- Draw 3 line segments from center }
    Sum := 0;
    for S := 0 to 2 do begin
      Sum := Sum + P[S];
      I := -PI / 2 + 2 * PI * Sum / 100.0;
      for R := 0 to 10 do begin
        X := Trunc(Cos(I) * R); Y := Trunc(Sin(I) * R);
        GotoXY (11 + X, 11 + Y); Write ('*');
        A[11 + X, 11 + Y] := 1;
      end;
    end;
  Ch := ReadKey; Sum := 0;
  { -- fill regions with letters }
  for S := 1 to 3 do begin
    LSum := Sum; Sum := Sum + P[S]; J := LSum;
    while J < Sum do begin
      I := -PI / 2 + 2 * PI * J / 100.0;
      for R := 1 to 9 do begin
        X := Trunc(Cos(I) * R); Y := Trunc(Sin(I) * R);
        if A[11 + X, 11 + Y] = 0 then begin
          GotoXY (11 + X, 11 + Y); Write (L[S]);
        end;
      end;
      Inc(J);
    end;
  end;
end.

```

```

{3.10}
program Thr10T91;
{ -- This program will convert large numbers in base 2,4,8,16. }
var
  A:      Array [1..255] of Byte;
  D:      String[1];
  NumSt:  String[65];
  I, J, K, L, M, N, X, Num, DigN,
  DigM, Pad, Ind, Pow, LInd, Zero, Sum: Byte;

begin
  Write ('Enter numeral: '); Readln (NumSt);
  Write ('Enter base M: ');  Readln (M);
  Write ('Enter base N: ');  Readln (N);
  L := Length (NumSt);
  DigM := Trunc (Ln (M) / Ln (2) + 0.001);
  DigN := Trunc (Ln (N) / Ln (2) + 0.001);
  Pad := DigN - (DigM * L mod DigN);
  if Pad = DigN then Pad := 0;
  for I := 1 to Pad do A[I] := 0;
  { -- Convert from base M to base 2 }
  for I := 1 to L do begin
    D := Copy (NumSt, I, 1);
    Num := Pos (D, '0123456789ABCDEF') - 1;
    for J := DigM - 1 downto 0 do begin
      Pow := 1;
      for K := 1 to J do Pow := Pow * 2;
      X := Num div Pow;
      Ind := I * DigM - J + Pad;
      A[Ind] := X;
      Num := Num - X * Pow;
    end;
  end;
  { -- Convert from base 2 to base N }
  LInd := DigM * L + Pad; Zero := 1;
  for I := 0 to (LInd div DigN) - 1 do begin
    Sum := 0;
    for J := 1 to DigN do begin
      Ind := I * DigN + J;
      Pow := 1;
      for K := 1 to (DigN - J) do Pow := Pow * 2;
      Sum := Sum + A[Ind] * Pow;
    end;
    if (Zero = 0) or (Sum > 0) then begin
      Zero := 0;
      Write (Copy ('0123456789ABCDEF', Sum + 1, 1));
    end;
  end;
end.

```