

```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '85 }
{ -- PASCAL PROGRAM SOLUTIONS }

{1.1}
program One1T85;
{ -- This program will simulate a Last-In-First-Out stack. }
var
  A: String [4];
  S: Integer;
  N: Array [1..9] of Integer;

begin
  S := 0;
  repeat
    Write ('Enter command: '); Readln (A);
    if A = 'ADD' then begin
      Inc(S);
      Write('Enter number: '); Readln (N[S]);
    end;
    if A = 'TAKE' then begin
      Writeln (N[S]); Dec(S);
    end;
  until A = 'QUIT';
end.
```



```
{1.2}
program One2T85;
{ -- This program will determine which number was erased. }
var
  N, S, I, T: Integer;
  Av: Real;

begin
  Write ('Enter N, AV: '); Readln (N, Av); S := 0;
  for I := 1 to N do S := S + I;
  for I := 1 to N do begin
    T := S - I;
    if T / (N - 1) = Av then begin
      Writeln ('NUMBER ERASED WAS ', I); Exit;
    end;
  end;
end.
```

```
{1.3}
program One3T85;
{ -- This program will print the square root of N. }
var
  D, I, T, V, Code: Integer;
  N, Q, S, Pow:      Real;
  A:                 String[9];
  C:                 Char;

begin
  Write ('Enter N, D: '); Readln (N, D);
  Q := Sqrt(N);  T := 0;
  Pow := 1;
  for I := 1 to Abs(D) do Pow := Pow * 10;
  if D < 0 then Pow := 1 / Pow;
  S := Int (Q / Pow + 0.5) * Pow;
  Str (S: 4:4, A);
  for I := 1 to Length(A) do begin
    C := A[I];
    if C <> '.' then begin
      Val(C, V, Code);  T := T + V;
    end;
  end;
  Writeln ('S=', S :9:4);
  Writeln ('SUM=', T :3);
end.
```

```
{1.4}
program One4T85;
{ -- This program will simulate a time dial. }
uses Crt;
var
  Y, J, K: Integer;

begin
  ClrScr;  Y := 1985;  J := 661;
  while Y <= 2345 do begin
    GotoXY (38,12);  Write (Y);
    if J > 10 then Dec(J,10);
    Delay (J);
    Inc(Y);
  end;
end.
```

```
{1.5}
program One5T85;
{ -- This program will determine # of tennis games and byes. }
var
  N, G, B, R, TG, BY: Integer;

begin
  Write ('Enter N: '); Readln (N);
  R := 0; TG := 0; BY := 0;
  while N > 1 do begin
    G := N div 2;
    if G * 2 = N then B := 0 else B := 1;
    Inc(R); Write ('ROUND ', R, ' ', G:2, ' GAMES');
    if B = 1 then
      Writeln (' 1 BYE')
    else
      Writeln;
    TG := TG + G; BY := BY + B; N := G + B;
  end;
  Writeln ('TOTAL ', TG:2, ' GAMES ', BY, ' BYES');
end.
```

```
{1.6}
program One6T85;
{ -- This program will find smallest, largest and sum of #s. }
var
  N, M, I, H, Num, T, U, L: Integer;
  S: LongInt;

begin
  Write ('Enter N, M: '); Readln (N, M); S := 0;
  if M > 999 then M := 999;
  if N < 100 then N := 100;
  for I := N to M do begin
    Num := I;
    H := Num div 100; Num := Num - H * 100;
    T := Num div 10; U := Num - T * 10;
    if (T = 0) or (U = 0) or (H = T) or (T = U) or (H = U) then
    else begin
      S := S + I; L := I;
      if S = I then Writeln ('SMALLEST = ', I);
    end;
  end;
  Writeln ('LARGEST = ', L);
  Writeln ('SUM = ', S);
end.
```

```
{1.7}
program One7T85;
{ -- This program will print a bill for Bob's Cycle shop. }
const
  A: Array [1..7] of String[4] =
    ('S193', 'S867', 'F234', 'S445', 'C492', 'J273', 'T100');
  B: Array [1..7] of String[20] =
    ('10 INCH SPROCKET', '30 INCH CHAIN', 'BLITZ MAG FRAME',
     'COMPUTCYCLE COMPUTER', 'JET BRAKE SET', '27 INCH WHEEL',
     '27X1 INCH TIRE TUBE');
  C: Array [1..7] of Real =
    (13.95, 27.50, 119.00, 33.95, 29.98, 32.00, 12.50);
var
  N, P:          String[10];
  I:            Integer;
  LT, LC, Tot, Tax: Real;

begin
  Write ('Enter Customer name: '); Readln (N);
  Write ('Enter part#: '); Readln (P);
  Write ('Enter labor time: '); Readln (LT);
  I := 1;
  while (P <> A[I]) and (I < 7) do Inc(I);
  Writeln ('CUSTOMER NAME: ', N);
  Writeln ('PART #: ', P);
  Writeln ('DESCRIPTION: ', B[I]);
  Writeln ('PART COST: ', C[I]: 6:2);
  LC := LT * 10;
  Writeln ('LABOR COST: ', LC: 6:2);
  Tax := C[I] * 0.05;
  Tax := Int(Tax * 100.0 + 0.501) / 100.0;
  Writeln ('5% TAX:      ', Tax :6:2);
  Tot := LC + C[I] + Tax;
  Writeln ('TOTAL:      ', Int(Tot * 100 + 0.5) / 100 :6:2);
end.
```

```
{1.8}
program One8T85;
{ -- This program will display labels alphabetically. }
const
  A: Array [1..6] of String[16] = ('LISA SPINXS', 'BOB SIMON',
    'BILL SIMON', 'HARRY TROUTMAN', 'HARRY PARKER', '*END*');
  B: Array [1..6] of String[8] = ('987-6543', '923-4455',
    '123-4567', '876-2174', '222-3333', '0');
var
  H, S, L, I, J: Integer;
  Rst, Lst:      String[10];
  X:             String[18];
  C:             Array [1..6] of String[18];
begin
  Write ('Enter # of lines on label: '); Readln (H);
  S := 1;
  while A[S] <> '*END*' do begin
    L := Length(A[S]); I := 1;
    while Copy(A[S], I, 1) <> ' ' do Inc(I);
    Rst := Copy(A[S], I+1, L-I); Lst := Copy (A[S], 1, I);
    C[S] := Rst + ', ' + Lst;
    Inc(S);
  end;
  Dec(S);
  for I := 1 to S - 1 do
    for J := I+1 to S do
      if C[I] > C[J] then begin
        X := C[I]; C[I] := C[J]; C[J] := X;
        X := B[I]; B[I] := B[J]; B[J] := X;
      end;
  for I := 1 to S do begin
    Writeln; Writeln (C[I]); Writeln (B[I]);
    for J := 1 to H - 3 do Writeln;
  end;
end.
```

```
{1.9}
program One9T85;
{ -- This program will guess secret letter in 5x5 matrix. }
uses Crt;
var
  I, J, S, X: Integer;
  C: Char;
  A: Array [0..24] of Integer;
  B: Array [1..5, 1..5] of Char;

begin
  ClrScr; Randomize; S := 11;
  for I := 0 to 24 do A[I] := 0;
  for I := 1 to 5 do
    for J := 1 to 5 do begin
      repeat
        X := Random(25);
      until A[X] = 0;
      B[I, J] := Chr(X + 65);
      GotoXY (13 + J * 2, I); Write (B[I, J]); A[X] := 1;
    end;

    I := 0; C := ' ';
    while (C <> 'Y') and (S > 0) do begin
      GotoXY (30, 2); Write ('SCORE=', S:2); Dec(S);
      GotoXY (10, 10); Inc(I);
      Write ('IS THE LETTER IN ROW ', I, ' '); Readln (C);
    end;
    J := 0; C := ' ';
    while (C <> 'Y') and (S > 0) do begin
      GotoXY (30, 2); Write ('SCORE=', S:2); Dec(S);
      GotoXY (10, 12); Inc(J);
      Write ('IS THE LETTER IN COL ', J, ' '); Readln (C);
    end;
    if S > 0 then Writeln ('YOUR LETTER IS ', B[I,J]);
  end.
```

```

{1.10}
program One10T85;
{ -- This program will display squares relative to cursor and #.
}
uses Crt;
var
  R, C, X, A, B: Integer;
  K: char;

begin
  ClrScr; R := 5; C := 5; K := ' ';
  while not (K in ['1' .. '4']) do begin
    GotoXY (C, R); Write ('#'); K := ' ';
    K := ReadKey;
    if K in ['I', 'J', 'K', 'M'] then begin
      GotoXY (C, R); Write (' ');
      if K = 'I' then Dec(R);
      if K = 'M' then Inc(R);
      if K = 'J' then Dec(C);
      if K = 'K' then Inc(C);
      K := '5';
    end;
  end;
  end;
  X := Ord(K) - Ord('0');
  if X = 1 then begin A := 1; B := 0; end;
  if X = 2 then begin A := 1; B := -1; end;
  if X = 3 then begin A := -1; B := -1; end;
  if X = 4 then begin A := -1; B := 0; end;
  if (R + 5*A > 24) or (R + 5*A < 1) or
     (C + 9*B > 80) or (C + 9*B < 1) then
    Writeln ('OFF THE SCREEN')
  else begin
    GotoXY (C + 8*B, R + 1*A); Writeln ('*****');
    GotoXY (C + 8*B, R + 2*A); Writeln ('*          *');
    GotoXY (C + 8*B, R + 3*A); Writeln ('*      ', X, '      *');
    GotoXY (C + 8*B, R + 4*A); Writeln ('*          *');
    GotoXY (C + 8*B, R + 5*A); Writeln ('*****');
  end;
end;
end.

```

```
{2.1}
program Two1T85;
{ -- This program will outline screen with random letters. }
uses Crt;
  var
    I, J, X: Integer;
    A, Ch:   Char;

begin
  repeat
    Randomize; ClrScr;
    for I := 1 to 11 do begin
      X := Random(26);  A := Chr(65 + X);
      GotoXY (I, I);
      for J := I to 80 - I do Write (A);
      for J := I+1 to 23-I do begin
        GotoXY (I, J);   Write (A);
        GotoXY (80-I, J); Write (A);
      end;
      GotoXY (I, 23-I);
      for J := I to 80 - I do Write (A);
      Ch := ReadKey;
    end;
  until Ch = Chr(27);
  ClrScr;
end.
```



```
{2.2}
program Two2T85;
{ -- This program will print the longest sequence of letters. }
var
  N, I, J, K: Integer;
  A:          Array [1..20] of Char;
  Found, One: Boolean;

begin
  Write ('Enter N: '); Readln (N);
  for I := 1 to N do begin
    Write ('Enter letter: '); Readln (A[I]);
  end;
  I := N; Found := False;
  while (I >= 2) and not Found do begin
    for J := 1 to N-I+1 do begin
      One := True;
      for K := 0 to I-2 do
        if A[J+K] >= A[J+K+1] then One := False;
      if One then begin
        for K := 0 to I-1 do Write (A[J+K], ' ');
        Writeln; Found := True;
      end;
    end;
    Dec(I);
  end;
end.
```

```
{2.3}
program Two3T85;
{ -- This program will change the margins for a given text. }
var
  A: String[128];
  W: String[20];
  C: Char;
  I, L, LW, LL: Integer;

begin
  Write ('Enter text: '); Readln (A); A := A + ' ';
  L := Length(A); LW := 5; Write (' ': 10); W := '';
  for I := 1 to L do begin
    C := A[I];
    if C <> ' ' then
      W := W + C
    else begin
      LL := Length(W);
      if LW + LL > 30 then begin
        Writeln; Write (' ': 5); LW := 0;
      end;
      if LL > 0 then begin
        Write (W, ' '); LW := LW + LL + 1; W := '';
      end;
      if (LL = 0) and (LW > 0) then begin
        Write (' '); Inc(LW);
      end;
    end;
  end;
end;
end.
```

```
{2.4}
program Two4T85;
{ -- This program will print word with consonants alphabetized. }
const
  Vowels: String[5] = 'AEIOU';
var
  I, J, L, VV, CC, VN, CN: Integer;
  A: String[20];
  B, X: Char;
  C: Array [1..20] of Char;
  V: Array [1..20] of Char;
  D: Array [1..20] of Char;

begin
  Write ('Enter word: '); Readln (A); L := Length(A);
  CN := 0; VN := 0; CC := 0; VV := 0;
  for I := 1 to L do begin
    B := A[I]; J := 1;
    while (J < 5) and (Copy(Vowels, J, 1) <> B) do Inc(J);
    if Copy (Vowels, J, 1) <> B then begin
      Inc(CN); C[CN] := B; D[I] := 'C'; end
    else begin
      Inc(VN); V[VN] := B; D[I] := 'V';
    end;
  end;

  { -- Sort Vowels }
  for I := 1 to VN-1 do
    for J := I+1 to VN do
      if V[I] > V[J] then begin
        X := V[I]; V[I] := V[J]; V[J] := X;
      end;

  { -- Sort Consonants }
  for I := 1 to CN-1 do
    for J := I+1 to CN do
      if C[I] > C[J] then begin
        X := C[I]; C[I] := C[J]; C[J] := X;
      end;

  for I := 1 to L do
    if D[I] = 'V' then begin
      Inc(VV); Write (V[VV]); end
    else begin
      Inc(CC); Write (C[CC]);
    end;
  Writeln;
end.
```

```
{2.5}
program Two5T85;
{ -- This program will print common letters and line up words. }
var
  N, I, J, K:      Integer;
  Common, Found: Boolean;
  X, Let:          Char;
  A:               Array [1..10] of String[15];

begin
  Write ('Enter N: '); Readln (N);
  for I := 1 to N do begin
    Write ('Enter word: '); Readln (A[I]);
  end;
  Found := False;
  for I := 1 to 26 do begin
    X := Chr(64 + I); Common := True; J := 1;
    while (J <= N) and Common do begin
      K := 1;
      while (K <= Length(A[J])) and (Copy(A[J], K, 1) <> X) do
        Inc(K);
      if Copy(A[J], K, 1) <> X then Common := False;
      Inc(J);
    end;
    if Common then begin
      Write (X, ' '); Found := True;
    end;
  end;
  if not found then begin
    Writeln ('NO COMMON LETTERS'); Exit;
  end;
  Writeln; Write ('Choose letter: '); Readln (Let);
  for I := 1 to N do begin
    J := 1;
    while (Copy(A[I], J, 1) <> Let) do Inc(J);
    Writeln (' ': 10 - J, A[I]);
  end;
end.
```

```
{2.6}
program Two6T85;
{ -- This program will keep score for a double dual race. }
var
  Init:              Array [1..21] of String[2];
  TeamName:          Array [1..3] of String[2];
  I, J, K:           Integer;
  StillUnique:       Boolean;
  UniqueTeams, Pl:   Integer;
  Team1Pos, Team2Pos: Array [1..7] of Integer;
  Team1, Team2:       Integer;
  Team1Pl, Team2Pl:  Integer;
```

```

begin
  UniqueTeams := 0;
  for I := 1 to 21 do begin
    Write ('Place ', I: 2, ': '); Readln (Init[I]);
    J := 0; StillUnique := True;
    while (J < UniqueTeams) and StillUnique and (I > 1) do begin
      Inc(J);
      if TeamName[J] = Init[I] then
        StillUnique := False;
    end; { -- while }
    if StillUnique then
      begin
        Inc(UniqueTeams);
        TeamName[UniqueTeams] := Init[I];
      end;
    end; { -- for I }
  { -- Assert that Team[1,2,3] = 3 unique team Initials. }

  for I := 1 to 2 do
    for J := I+1 to 3 do begin
      PL := 0; Team1 := 0; Team2 := 0;
      Team1Pl := 0; Team2Pl := 0;
      for K := 1 to 21 do begin
        if Init[K] = TeamName[I] then
          begin
            Inc(Pl);
            Team1 := Team1 + Pl;
            Inc(Team1Pl);
            Team1Pos[Team1Pl] := Pl
          end;
        if Init[K] = TeamName[J] then
          begin
            Inc(Pl);
            Team2 := Team2 + Pl;
            Inc(Team2Pl);
            Team2Pos[Team2Pl] := Pl
          end;
      end; { -- for K }
      Team1 := Team1 - Team1Pos[6] - Team1Pos[7];
      Team2 := Team2 - Team2Pos[6] - Team2Pos[7];
      Writeln ('TEAM ', TeamName[I], ': ', Team1, ' POINTS');
      Writeln ('TEAM ', TeamName[J], ': ', Team2, ' POINTS');
      if (Team1 < Team2)
      or ((Team1 = Team2) and (Team1Pos[6] < Team2Pos[6])) then
        Write ('TEAM ', TeamName[I])
      else
        Write ('TEAM ', TeamName[J]);
      Writeln (' WINS!'); Writeln;
    end; { -- for J }
  end.

```

```
{2.7}
program Two7T85;
{ -- This program will allow manipulation of 3x3 array of data. }
uses Crt;
var
  A: Array [1..4, 1..4] of Real;
  Tot: Real;
  I, J, Row, Col: Integer;
  C, Ch: Char;

begin
  A[1,1] := 10.11;  A[1,2] := 20.22;  A[1,3] := 30.33;
  A[2,1] := 11.1;   A[2,2] := 22.2;   A[2,3] := 33.3;
  A[3,1] := 10.0;   A[3,2] := 20.0;   A[3,3] := 30.0;
  C := ' ';
  while C <> 'C' do begin
    ClrScr;
    Writeln ('A. EDIT OR CHANGE A VALUE');
    Writeln ('B. DISPLAY THE RESULTS');
    Writeln ('C. QUIT');
    Write ('Enter option: ');  Readln (C);
    if C = 'A' then begin
      Write ('Enter row, col: ');  Readln (Row, Col);
      Write ('Enter number: ');    Readln (A[Row, Col]);
    end
    else if C = 'B' then begin
      for I := 1 to 3 do A[I, 4] := 0;
      for J := 1 to 3 do A[4, J] := 0;
      Tot := 0;
      for I := 1 to 3 do begin
        for J := 1 to 3 do begin
          Write (A[I,J] :6:2, ' ');  Tot := Tot + A[I, J];
          A[4, J] := A[4, J] + A[I, J];
          A[I, 4] := A[I, 4] + A[I, J];
        end;
        Writeln (A[I, 4]: 6:2);
      end;
      for J := 1 to 3 do Write (A[4, J] :6:2, ' ');
      Write (Tot :6:2);
    end;
    if C <> 'C' then begin
      Writeln;  Write ('Press any key: ');  Ch := ReadKey;
    end;
  end; { -- while }
end.
```

```
{2.8}
program Two8T85;
{ -- This program will print all combinations of 4 digits. }
var
  A, B, C, D, P, S, Code: Integer;
  Pst: String[2];

begin
  S := 0;
  for A := 1 to 8 do
    for B := A+1 to 9 do begin
      P := A * B;
      if P >= 10 then begin
        Str(P, Pst);
        Val(Copy(Pst,1,1), C, Code);
        Val(Copy(Pst,2,1), D, Code);
        if (A <> C) and (A <> D) and (B <> C) and (B <> D) then
          begin
            Write (A, ' ', B, ' ', C, ' ', D, ' ');
            Writeln (A, ' X ', B, ' = ', P);
            Inc(S);
          end;
        end;
      end;
    end;
  Writeln ('TOTAL = ', S);
end.
```

```
{2.9}
program Two9T85;
{ -- This program will select words given a string w/ wildcard. }
var
  A:          Array[1..25] of String[11];
  I, J, N, L, W: Integer;
  St, X, Ri, Le: String[11];

begin
  Write ('Enter N: '); Readln (N);
  for I := 1 to N do begin
    Write ('Enter word: '); Readln (A[I]);
  end;
  repeat
    Write ('Enter string: '); Readln (St);
    L := Length(St); W := 0; X := '';
    I := Pos('*', St);
    if I = 0 then Exit;
    { -- Asterisk is at position I }
    { -- Compare Left part of string and Right part of string. }
    Le := Copy(St, 1, I-1); Ri := Copy (St, I+1, L-I);
    for J := 1 to N do
      if (Copy(A[J], 1, I-1) = Le) and
        (Copy(A[J], Length(A[J]) - (L-I) + 1, L-I) = Ri) then
        begin
          Writeln (A[J]); W := 1;
        end;

    if W = 0 then Writeln ('NO WORDS FOUND'); Writeln;
  until I = 0;
end.
```



```

{2.10}
program Two10T85;
{ -- This program will maintain air conditioning in 3 rooms. }
uses Crt;
var
  Off, Co, Dr:      Real;
  S, M, O, C, D, Ch, Air, LM: Integer;
  OfAir, CoAir, DrAir:      Real;

begin
  Write ('Enter last 5-minutes: '); Readln (LM);
  ClrScr;
  Off := 72;      Co := 65;      Dr := 79;
  OfAir := 0;    CoAir := 0;    DrAir := 0;
  S := 0;        M := 0;        Ch := 0;
  O := 0;        C := 0;        D := 0;
  Writeln ('OF CO DS OFFICE COMP. DRY. MIN:SEC');
  repeat
    if ((M mod 5 = 0) and (S = 0)) or (Ch = 1) then begin
      Write (O, ' ', C, ' ', D, ' ');
      Write (Off: 3:1, ' ', Co :3:1, ' ', Dr :3:1);
      Write (' ', M:3, ':');
      if S > 0 then Writeln (S) else Writeln ('00');
      Ch := 0;
    end;
    S := S + 15;
    if S = 60 then begin
      Inc(M); S := 0;
    end;
    Off := Off + 0.1 - OfAir;
    Co := Co + 0.2 - CoAir;
    Dr := Dr + 0.1/4 - DrAir;
    if (Off > 78) and (O = 0) then begin O := 1; Ch := 1; end;
    if (Co > 70) and (C = 0) then begin C := 1; Ch := 1; end;
    if (Dr > 85) and (D = 0) then begin D := 1; Ch := 1; end;
    if (Off < 72) and (O = 1) then begin O := 0; Ch := 1; end;
    if (Co < 65) and (C = 1) then begin C := 0; Ch := 1; end;
    if (Dr < 75) and (D = 1) then begin D := 0; Ch := 1; end;
    Air := (O + C + D) * 2;
    if Air = 0 then begin
      OfAir := 0; CoAir := 0; DrAir := 0; end
    else begin
      OfAir := O / Air; CoAir := C / Air; DrAir := D / Air;
    end;
  until (M = LM) and (S > 0);
end.

```

```
{3.1}
program Thr1T85;
{ -- This program will display the sides of a die. }
{ -- 6 ways to represent die (each with different top)
  DATA Top, Front, Right, Back, Left, (Bottom derived) }
const
  A: Array[1..30] of Integer =
    (1, 5, 4, 2, 3, 6, 5, 3, 2, 4, 5, 1, 3, 6, 4,
     2, 1, 4, 6, 3, 3, 5, 1, 2, 6, 4, 5, 6, 2, 1);
var
  T, F, I, J, R: Integer;

begin
  Write ('Enter Top, Front: '); Readln (T, F);
  { -- Determine which data set of 5 to use (based on top #) }
  I := 1;
  while A[I] <> T do I := I + 5;
  { -- Rotate sides till a side matches the front # }
  J := 1;
  while (A[I + J] <> F) do J := J + 1;
  if J = 4 then J := 0;
  R := J + 1;
  { -- Generate rest of sides, sum of opposites sides = 7 }
  Writeln ('TOP = ', T, ' FRONT = ', F, ' RIGHT = ', A[I+R]);
  Write ('BACK = ', 7-F, ' LEFT = ', 7 - A[I+R]);
  Writeln (' BOTTOM = ', 7-T);
end.
```

```
{3.2}
program Thr2T85;
{ -- This program will factor a quadratic equation. }
var
  A, B, C, D, E, H, I, K, N, S: Integer;
  R: Array [1..2] of Integer;
  Displayed: Boolean;

begin
  Write ('Enter A, B, C: '); Readln (A, B, C);
  if A < 0 then begin
    A := -A; B := -B; C := -C;
  end;
  if A > 1 then
    for I := A downto 2 do
      if (A mod I = 0) and (B mod I = 0) and (C mod I = 0) then
        begin
          A := A div I; B := B div I; C := C div I; Write (I);
        end;

  S := B * B - 4 * A * C;
  if S < 0 then begin
    Writeln ('CANNOT BE FACTORED'); Exit;
  end;
  H := Trunc (Sqrt(S) + 0.01); E := 2 * A;
  R[1] := -B + H; R[2] := -B - H;
  for K := 1 to 2 do begin
    D := E; N := R[K]; I := D; Displayed := False;
    repeat
      if (N mod I = 0) and (D mod I = 0) then begin
        N := N div I; D := D div I;
        Write ('(');
        if D > 1 then Write (D);
        Write ('X');
        if N < 0 then Write ('+', (-N), ')');
        if N > 0 then Write ('-', N, ')');
        Displayed := True;
      end;
      Dec(I);
    until Displayed;
  end;
end.
```

```
{3.3}
program Thr3T85;
{ -- This program will simulate a calculator. }
var
  I, J, K, L, Code: Integer;
  Ex, C: String[20];
  Ch: String[1];
  S: Real;
  B: Array [1..10] of Integer;
  A: Array [1..10] of Real;

begin
  Write ('Enter expression: '); Readln (Ex);
  L := Length(Ex); C := ''; J := 0;
  for I := 1 to L do begin
    Ch := Copy (Ex, I, 1);
    if Ch >= '0' then
      C := C + Ch
    else begin
      Inc(J); Val(C, A[J], Code); C := '';
      B[J] := Pos(Ch, '+-*/');
    end;
  end;
  Inc(J); Val(C, A[J], Code); K := 1;
  for I := 1 to J-1 do
    if B[I] < 3 then begin
      B[K] := B[I]; Inc(K); A[K] := A[I+1]; end
    else
      if B[I] = 3 then
        A[K] := A[K] * A[I+1]
      else { -- B = 4 }
        A[K] := A[K] / A[I+1];

  S := A[1];
  for I := 1 to K-1 do
    if B[I] = 2 then S := S - A[I+1] else S := S + A[I+1];
  Writeln (S: 7:3);
end.
```

```

{3.4}
program Thr4T85;
{ -- This program will compute all digits of N factorial. }
var
  N, I, J, D, C, CC: Integer;
  A: Array [1..254] of Integer;

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

```

```

{3.5}
program Thr5T85;
{ -- This program will sum and subtract 2 big decimals. }
var
  Ast, Bst: String[31];
  A, B, C, D: Array [1..30] of Integer;
  I, J, LenA, LenB, X, S, G,
  H, Y, Z, L, Code, Car, Bor: Integer;

begin
  Write ('Enter #1: '); Readln (Ast); LenA := Length(Ast);
  Write ('Enter #2: '); Readln (Bst); LenB := Length(Bst);
  S := 0;
  for I := LenA downto 1 do
    if Copy(Ast, I, 1) = '.' then
      X := I
    else begin
      Inc(S); Val(Copy(Ast, I, 1), A[S], Code);
    end;
  S := 0;
  for I := LenB downto 1 do
    if Copy (Bst, I, 1) = '.' then
      Y := I
    else begin
      Inc(S); Val(Copy(Bst, I, 1), B[S], Code);
    end;
  end;

```

```

{ -- Align decimal point }
G := LenA - X;  H := LenB - Y;
if G > H then L := G else L := H;
Z := G - H;
if Z > 0 then { -- Second # is smaller, so place leading 0s. }
  begin
    for I := LenB-1 downto 1 do begin
      B[I+Z] := B[I];  B[I] := 0;
    end;
    LenB := LenB + Z;
  end;
if Z < 0 then { -- First # is smaller, so put leading 0s. }
  begin
    for I := LenA-1 downto 1 do begin
      A[I-Z] := A[I];  A[I] := 0;
    end;
    LenA := LenA - Z;
  end;

if LenA > LenB then Y := LenA - 1 else Y := LenB - 1;
Car := 0;  Bor := 0;
{ -- Add and subtract }
for I := 1 to Y do begin
  C[I] := A[I] + B[I] + Car;  Car := C[I] div 10;
  C[I] := C[I] - Car * 10;
  D[I] := A[I] - B[I] - Bor;
  if D[I] < 0 then Bor := 1 else Bor := 0;
  D[I] := D[I] + Bor * 10;
end;
Write ('SUM = ');
if Car > 0 then Write (Car);
for I := Y downto 1 do begin
  if I = L then Write ('.');
  Write (C[I]);
end;
Writeln;  Write ('DIFFERENCE = ');
for I := Y downto 1 do begin
  if I = L then Write ('.');
  Write (D[I]);
end;
end.

```

```

{3.6}
program Thr6T85;
{ -- This program will control the movements of a snake. }
uses Crt;
const
  SnakeLen = 30;
var
  V, H, I, X, Y: Integer;
  VCoord, HCoord: Array [1..SnakeLen] of Integer;
  FrontHV, EndHV: Integer;
  Ch: Char;
  InvalidKey: Boolean;

```

```

begin
  ClrScr;
  InvalidKey := False;
  V := 12;  H := 40-(SnakeLen div 2);  GotoXY (H,V);
  FrontHV := 0;  EndHV := 1;
  { -- Center snake (asterisks) on the screen }
  for I := H to (H + SnakeLen - 1) do begin
    Write ('*');
    Inc(FrontHV);
    VCoord[FrontHV] := V;
    HCoord[FrontHV] := I;
  end;
  Ch := ReadKey;

  repeat
    H := HCoord[FrontHV];
    V := VCoord[FrontHV];
    for I := 1 to 2000 do
      If KeyPressed then Ch := ReadKey;

      case Ch of
        'I', 'i' : Dec(V);
        'M', 'm' : Inc(V);
        'J', 'j' : Dec(H);
        'K', 'k' : Inc(H);
      end;

      for I := 1 to SnakeLen do
        if (H = HCoord[I]) and (V = VCoord[I]) then
          InValidKey := True;

      if InValidKey or (V = 0) or (V = 25) or (H = 0) or (H = 80)
then
      InvalidKey := True
    else begin
      GotoXY (H,V); Write ('*');
      Y := HCoord[EndHV];
      X := VCoord[EndHV];
      GotoXY (Y,X); Write (' ');
      HCoord[EndHV] := H;
      VCoord[EndHV] := V;
      Inc(FrontHV);
      if FrontHV > SnakeLen then
        FrontHV := 1;
      Inc(EndHV);
      If EndHV > SnakeLen then
        EndHV := 1;
      end; { -- else }
    until InvalidKey;
  end.

```

```
{3.7}
program Thr7T85;
{ -- This program will print 3 permutations of a word. }
var
  A:   String[8];
  Let: Char;
  F:   Array [1..7] of Integer;
  B:   Array [1..7] of Byte;
  KK, L, I, J, Fac, T, S, K, X: Integer;

begin
  Write ('Enter word: '); Readln (A); L := Length(A);
  Write ('Enter K: ');     Readln (KK);
  { -- Alphabetize letters }
  for I := 1 to L-1 do
    for J := I+1 to L do
      if A[I] > A[J] then begin
        Let := A[I]; A[I] := A[J]; A[J] := Let;
      end;

  { -- Produce factorials F(I) = (I-1)! }
  for I := 1 to L do begin
    Fac := 1;
    for J := 1 to I-1 do Fac := Fac * J;
    F[I] := Fac;
  end;

  for T := 1 to 3 do begin
    K := KK * T - 1;
    for I := 1 to L do B[I] := 0;
    { -- Generate Kth permutation }
    for I := L downto 1 do begin
      X := K div F[I]; S := 0; J := 1;
      repeat
        while B[J] > 0 do Inc(J);
        Inc(S);
        if S > X then begin
          B[J] := 1; Write (A[J]); end
        else
          Inc(J);
      until (J > L) or (S > X);
      K := K - F[I] * X;
    end;
    Write (' ');
  end;
  Writeln;
end.
```



```
{3.8}
program Thr8T85;
{ -- This program will display N pennies on board. }
uses Crt;
var
  N, Sp, J, S, I:      Integer;
  A:  Array [1..14] of Integer;
  Ch: Char;

begin
  Write ('Enter N: '); Readln (N);
  ClrScr; Writeln ('TOTAL = ', N);
  if N = 8 then Sp := 1; { -- 8 and 14 are special cases }
  J := N mod 2;  J := 2 - J;  S := J;
  if N = 14 then S := J + 2;
  Write (' ');
  for I := 1 to N do begin
    Write (I mod 10 :2);
  end;
  Writeln;
  for I := 1 to N do Writeln (I mod 10);
  for I := 1 to N do begin
    A[I] := S;
    if (N = 14) and (I = 14) then begin
      S := 2;  A[I] := S;
    end;
    GotoXY (2*S+1, 2+I); Write ('*');
    S := S + 2 + Sp;
    if S > N then
      if (Sp = 1) then S := S - N else S := (N mod 2) + 1;
  end;
  Ch := ReadKey;
  for I := 1 to N do begin
    GotoXY (45, I+2); Write ('(', I, ', ', A[I], ')');
    Writeln ('  SUM = ', I + A[I]);
  end;
end.
```

```
{3.9}
program Thr9T85;
{ -- This program will determine # of moves made to a stack. }
var
  N, I: Integer;
  A: Array [1..15] of Integer;

begin
  { 1 block - 1 move (obvious)
    2 blocks- 3 moves (Move 1 stack, move #2, move 1 stack)
    3 blocks- 7 moves (Move 2 stack, move #3, move 2 stack on #3)
                    (3 moves      + 1 move + 3 moves)
    4 blocks-15 moves (Move 3 stack, move #4, move 3 stack on #4)
                    (7 moves      + 1 move + 7 moves) }
  Write ('Enter N: '); Readln (N);
  A[1] := 1;
  for I := 2 to N do A[I] := A[I-1] * 2 + 1;
  Writeln (A[N])
end.
```

```

{3.10}
program Thr10T85;
{ -- This program will find sets of #s P, Q, R (P = Q x R). }
var
  S, I, J, NU, X1, X2, Y1, Y2, Z2: Integer;
  X, C, Code: Integer;
  Dupl: Boolean;
  Prod, Q, R: LongInt;
  P: String[5];
  A: Array [0..9] of Integer;

begin
  Write ('Enter S: '); Readln (S);
  Q := S;
  repeat
    repeat
      Inc(Q); X1 := Q div 10; Y1 := Q mod 10;
    until X1 <> Y1;
    NU := 10000 div Q;
    for R := NU to 999 do begin
      Dupl := False;
      for I := 0 to 9 do A[I] := 0;
      X2 := R div 100; C := R - X2 * 100;
      Y2 := C div 10; Z2 := C - Y2 * 10;
      if (X2 <> Y2) and (Y2 <> Z2) and (X2 <> Z2) and
        (X1 <> X2) and (X1 <> Y2) and (X1 <> Z2) and
        (Y1 <> X2) and (Y1 <> Y2) and (Y1 <> Z2) then
        begin
          A[X1] := 1; A[Y1] := 1; A[X2] := 1;
          A[Y2] := 1; A[Z2] := 1;
          Prod := Q * R;
          Str (Prod, P);
          if Length(P) = 5 then begin
            for I := 1 to 5 do begin
              Val(Copy(P, I, 1), X, Code);
              if A[X] = 1 then Dupl := True;
            end;
            for I := 1 to 4 do
              for J := I+1 to 5 do
                if Copy(P, I, 1) = Copy(P, J, 1) then Dupl := True;
            if not Dupl then begin
              Writeln ('P = ', P, ' Q = ', Q, ' R = ', R);
            end;
          end;
        end; { -- if }
      end; { -- if }
    end; { -- for }
  until Q > 99;
end.

```