```
{ -- FLORIDA HIGH SCHOOLS COMPUTING COMPETITION '92 }
{ -- PACSCAL PROGRAM SOLUTIONS }


{1.1}
program One1T92;
{ -- This program displays the company name: GTEDS. }

begin
  Writeln ('GGGGG   TTTTT   EEEEE');
  Writeln ('G         T     E');
  Writeln ('G GGG     T     EEEEE   DATA SERVICES');
  Writeln ('G   G     T     E');
  Writeln ('GGGGG     T     EEEEE');
end.


{1.2}
program One2T92;
{ -- This program will display the company name in a year. }
  var
    Year: Integer;

begin
  Write ('Enter year: ');  Readln (Year);
  if Year < 1920 then
    Writeln ('RICHLAND CENTER TELEPHONE COMPANY')
  else if Year < 1926 then
    Writeln ('COMMONWEALTH TELEPHONE COMPANY')
  else if Year < 1935 then
    Writeln ('ASSOCIATED TELEPHONE UTILITIES COMPANY')
  else if Year < 1959 then
    Writeln ('GENERAL TELEPHONE CORPORATION')
  else if Year < 1982 then
    Writeln ('GENERAL TELEPHONE & ELECTRONICS CORPORATION')
  else
    Writeln ('GTE CORPORATION');
end.


{1.3}
program One3T92;
{ -- This program will determine company's ranking in Forbes. }
  var
    Rank, Places: Integer;

begin
  Write ('Enter 1991 rank: ');  Readln (Rank);
  Write ('Enter number of places: ');  Readln (Places);
  Writeln (Rank - Places);
end.
```

```
{1.4}
program One4T92;
{ -- This program will indent GTE's 6 operations. }
  var
    X: Byte;
begin
  Write ('Enter number of spaces: ');  Readln (X);
  Writeln ('GTE TELEPHONE OPERATIONS');
  Writeln (' ': X, 'GTE GOVERNMENT SYSTEMS');
  Writeln (' ': X * 2, 'GTE MOBILE COMMUNICATIONS');
  Writeln (' ': X * 3, 'GTE INFORMATION SERVICES');
  Writeln (' ': X * 4, 'GTE SPACENET');
  Writeln (' ': X * 5, 'GTE AIRFONE');
end.


{1.5}
program One5T92;
{ -- This program will display # of WHOLE YEARS GTEDS existed. }
  var
    Month, Year, X: Integer;

begin
  Write ('Enter month, year: ');  Readln (Month, Year);
  if Month < 10 then
    X := 1
  else
    X := 0;
  Writeln (Year - 1967 - X, ' YEARS');
end.


{1.6}
program One6T92;
{ -- This program will center a title and name in a box. }
  var
    Title, Name:    String[20];
    I, L, Sp1, Sp2: Byte;

begin
  Write ('Enter title: ');  Readln (Title);
  Write ('Enter name: ');  Readln (Name);
  for I := 1 to 24 do Write ('*');
  Writeln;
  Writeln ('*', ' ': 22, '*');

  L := Length(Title) + Length(Name) + 1;
  Sp1 := (22 - L) div 2;
  Sp2 := (22 - L) - Sp1;
  Writeln ('*', ' ': Sp1, Title, ' ', Name, ' ': Sp2, '*');

  Writeln ('*', ' ': 22, '*');
  for I := 1 to 24 do Write ('*');
end.
```

```
{1.7}
program One7T92;
{ -- This program will display a 4-line statement for ISOP. }
  var
    Name, Title, Group: String[25];

begin
  Write ('Enter name: ');  Readln (Name);
  Write ('Enter title: '); Readln (Title);
  Write ('Enter group: '); Readln (Group);
  Writeln (Name, ' IS A ', Title, ' WITHIN THE');
  Writeln (Group, ' GROUP AND');
  Writeln ('HAS BEEN SELECTED TO PARTICIPATE IN');
  Writeln ('THE ISOP.');
end.


{1.8}
program One8T92;
{ -- This program will display a dollar sign next to an amount. }
  var
    St:     String[7];
    Code:   Integer;
    Amount: Real;

begin
  Write ('Enter amount: ');  Readln (St);
  Val(St, Amount, Code);
  if Amount >= 2000 then
    Writeln ('$2000.00')
  else
    Writeln ('$', St);
end.


{1.9}
program One9T92;
{ -- This program will display an acronym for business words. }
  var
    St: String[80];
    I:  Byte;

begin
  Write ('Enter words: ');  Readln (St);
  Write (Copy(St, 1, 1));
  for I := 2 to Length(St) - 1 do
    if Copy(St, I , 1) = ' ' then
      Write (Copy(St, I + 1, 1));
end.
```

```
{1.10}
program One10T92;
{ -- This program will calculate QUALITY hours and minutes. }
  var
    N, M, Total, Hours, Min: LongInt;

begin
  Write ('Enter number of technicians, N: '); Readln (N);
  Write ('Enter number of minutes, M: ');     Readln (M);
  Total := 50 * 5 * N * M;
  Hours := Total div 60;
  Min   := Total - Hours * 60;
  Writeln (Hours, ' HOURS ', Min, ' MINUTES');
end.
```

```
{2.1}
program Two1T92;
{ -- This program will display a speech indented. }
  var
    Line: Array [1..10] of String[40];
    I, J: Byte;
    Ch:   Char;

begin
  I := 0;
  repeat
    Inc(I);
    Write ('Enter Line: ');  Readln (Line[I]);
  until Line[I] = '';
  for J := 1 to I - 1 do begin
    Ch := Line[J,1];
    if Ch = 'I' then
      Writeln (Line[J])
    else if Ch in ['A' .. 'H'] then
      Writeln (' ': 4, Line[J])
    else
      Writeln (' ': 8, Line[J]);
  end;
end.


{2.2}
program Two2T92;
{ -- This program will display a number in words. }
  const
    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
    Num, Units, Tens: Byte;

begin
  Write ('Enter number: ');  Readln (Num);
  if Num < 20 then
    Writeln (Words[Num])
  else begin
    Tens  := Num div 10;
    Units := Num - Tens * 10;
    Write (Words[18 + Tens]);
    if Units > 0 then
      Write ('-', Words[Units]);
  end;
end.
```

```
{2.3}
program Two3T92;
{ -- This program will display selected items from a NRD menu. }
uses Crt;
  const
    Crit: Array [1..7] of String[50] =
      ('DEMONSTRATED INTEREST IN INFORMATION MANAGEMENT.',
       'DEMONSTRATED LEADERSHIP SKILLS.',
       'STRONG GPA/PERFORMANCE HISTORY.',
       'AT LEAST TWO COURSES IN ANY PROGRAMMING LANGUAGE.',
       'INTERNSHIP OR WORK EXPERIENCE.',
       'EFFECTIVE ORAL AND WRITTEN COMMUNICATION SKILLS.',
       'CAREER DEVELOPMENT POTENTIAL.');
  var
    Name, Degree, Items: String[40];
    I, Num: Byte;
    Ist:    String[1];

begin
  Write ('Enter name: ');    Readln (Name);
  Write ('Enter degree: ');  Readln (Degree);
  for I := 1 to 7 do Writeln (I, '. ', Crit[I]);
  Writeln;
  Write ('Select up to 7 items: ');  Readln (Items);

  ClrScr;
  Writeln (Name);  Writeln (Degree);
  Num := 0;
  for I := 1 to 7 do begin
    Str(I, Ist);
    if Pos(Ist,Items) > 0 then begin
      Inc(Num);
      Writeln;  Writeln (Num, '. ', Crit[I]);
    end;
  end;
end.
```

```
{2.4}
program Two4T92;
{ -- This program will rate a speech. }
  const
    Cat: Array [1..7] of String[16] =
        ('SPEECH VALUE', 'PREPARATION', 'MANNER', 'ORGANIZATION',
         'OPENING', 'BODY OF SPEECH', 'CONCLUSION');
    Verbal: Array [1..5] of String[15] =
        ('EXCELLENT', 'ABOVE AVERAGE', 'SATISFACTORY',
         'SHOULD IMPROVE', 'MUST IMPROVE');
  var
    Rating:        Array [1..7] of String[15];
    I, Num, Total: Byte;
    Ave:           Real;

begin
  for I := 1 to 7 do begin
    Write ('Enter rating for ', Cat[I], ': ');
    Readln (Rating[I]);
  end;
  Total := 0;
  for I := 1 to 7 do begin
    Num := 1;
    while (Rating[I] <> Verbal[Num]) and (Num < 7) do
      Inc(Num);
    Writeln (Cat[I], ': ', Num);
    Inc(Total, Num);
  end;
  Writeln;
  Ave := Total / 7.0;
  Writeln ('AVERAGE NUMERICAL RATING = ', Ave: 2:1);
  Writeln ('SPEECH RATING = ', Verbal[Round(Ave)]);
end.
```

```
{2.5}
program Two5T92;
{ -- This program will format GTEDS MISSION statement. }
  const
    St: Array [1..4] of String[55] =
      ('BE THE CUSTOMER-ORIENTED LEADER AND PROVIDER-OF-CHOICE ',
       'OF QUALITY INFORMATION PRODUCTS AND SERVICES IN THE ',
       'TELECOMMUNICATIONS MARKETPLACE AND SELECTED OTHER ',
       'RELATED MARKETS IN SUPPORT OF GTE''S TELOPS GOALS.');
   var
     State: String[220];
     Line: String[40];
     Word: String[20];
     Ch:   Char;
     NumCh, I, J, N: Byte;

begin
  Write ('Enter N: ');  Readln (N);
  State := St[1] + St[2] + St[3] + St[4];
  Word  := '';  Line := '';
  for I := 1 to Length(State) do begin
    Ch := State[I];
    Word := Word + Ch;
    if Ch in [' ', '-', '.'] then begin
      NumCh := Length(Line) + Length(Word);
      if Ch = ' ' then Dec(NumCh);
      if NumCh > N then
        begin
          Writeln (Line);
          Line := Word;
        end
      else
        Line := Line + Word;
      Word := '';
    end;
  end;  { -- for I }
  Writeln (Line + Word);
end.
```

```
{2.6}
program Two6T92;
{ -- This program will change (.) to (?) at end of sentence. }
  const
    Quest: Array[1..5] of String[5] =
         ('WHAT', 'WHY', 'HOW', 'WHO', 'WHERE');
  var
    Par:       String[255];
    FirstW:    String[20];
    FirstWord: Boolean;
    Ch:        Char;
    I, J:      Byte;

begin
  Write ('Enter paragraph: ');  Readln (Par);  Writeln;
  FirstW := '';  FirstWord := True;
  for I := 1 to Length(Par) do begin
    Ch := Par[I];
    if (Ch = ' ') and (Length(FirstW) > 0) then
      FirstWord := False
    else if Ch in ['.', '!', '?'] then
      begin
        if Ch = '.' then
          for J := 1 to 5 do
            if FirstW = Quest[J] then Ch := '?';
        FirstW := '';  FirstWord := True;
      end
    else if FirstWord and (Ch <> ' ') then
      FirstW := FirstW + Ch;
    Write(Ch);
  end;
end.
```

```
{2.7}
program Two7T92;
{ -- This program will print names in the office at a beep. }
  const
    Name: Array[1..14] of String[10] = ('DAVID', 'DON', 'DOUG',
         'GRANDVILLE', 'JAMES', 'JIM', 'JOHN', 'LINDA', 'MARIE',
         'MATT', 'PAULA', 'ROBERT', 'SHELLEY', 'TOM');
    Start: Array[1..14] of Integer = (0700, 0800, 0730, 1230,
          1130, 0900, 0700, 1230, 0700, 1230, 0700, 0800,
          0630, 1100);
    Quit: Array[1..14] of Integer = (1600, 1700, 1630, 2100,
          2200, 1800, 1600, 2300, 1600, 2300, 1600, 1700,
          1530, 1930);
   var
     Time, I, Num: Integer;
     Day:          String[10];
     InOffice:     Boolean;

begin
  Write ('Enter time: ');  Readln (Time);
  Write ('Enter day: ');   Readln (Day);
  Num := 0;
  for I := 1 to 14 do begin
    if (Start[I] <= Time) and (Time <= Quit[I]) then
      if (Day <> 'SUNDAY') and (Day <> 'SATURDAY') then begin
        InOffice := True;
        if (Name[I] = 'JAMES') and (Day = 'MONDAY') then
          InOffice := False;
        if (Name[I] = 'LINDA') and (Day = 'FRIDAY') then
          InOffice := False;
        if (Name[I] = 'MATT')  and (Day = 'MONDAY') then
          InOffice := False;
        if InOffice then begin
          Inc(Num);
          if Num = 1 then
            Write (Name[I])
          else
            Write (', ', Name[I]);
        end;
      end;
  end;  { -- for I }
  if Num = 0 then Writeln ('NONE');
end.
```

```
{2.8}
program Two8T92;
{ -- This program will randomly assign titles to a team. }
  const
    Name: Array[1..7] of String[7] = ('WILL', 'DARLENE',
          'JEFF', 'LIZ', 'LORI', 'MARY', 'PING');
    Title: Array[1..5] of String[9] = ('AUTHOR',
          'MODERATOR', 'READER', 'RECORDER', 'INSPECTOR');
  var
    I, J, X: Byte;
    TName:   Array[1..5] of String[7];
    Valid:   Boolean;

begin
  Randomize;
  Write ('Enter author''s name: ');  Readln (TName[1]);

  { -- Choose a moderator: TName[2] }
  if TName[1] = Name[1] then
    TName[2] := Name[2]
  else if TName[1] = Name[2] then
    TName[2] := Name[1]
  else
    TName[2] := Name[Random(2) + 1];

  { -- Choose next 3 title names. }
  for I := 3 to 5 do begin
    repeat
      Valid := True;
      X := Random(7) + 1;
      for J := 1 to I do
        if Name[X] = TName[J] then Valid := False;
    until Valid;
    TName[I] := Name[X];
  end;
  { -- Display all 5 titles and names. }
  for I := 1 to 5 do
    Writeln (Title[I], ' - ', TName[I]);
end.
```

```
{2.9}
program Two9T92;
{ -- This program will sort a list of names with area codes. }
  var
    Name: Array[1..15] of String[15];
    I, J, A, Area1, Area2, Num: Integer;
    X: String[15];

begin
  Write ('Enter two area codes: ');  Readln (Area1, Area2);
  Write ('Enter number of names: '); Readln (Num);
  for I := 1 to Num do begin
    Write ('Enter name: ');  Readln (Name[I]);
  end;
  for I := 1 to Num - 1 do
    for J := I + 1 to Num do
      if Name[I] > Name[J] then begin
        X := Name[I];  Name[I] := Name[J];  Name[J] := X;
      end;

  if Area1 > Area2 then begin
    A := Area1;  Area1 := Area2;  Area2 := A;
  end;

  for I := 1 to (Num + 1) div 2 do
    Writeln (Area1, ' - ', Name[I]);
  for I := (Num + 1) div 2 + 1 to Num do
    Writeln (Area2, ' - ', Name[I]);
end.



{2.10}
program Two10T92;
{ -- This program will adjust a golf score by handicap. }
  const
    Par: Array[1..9] of Byte = (5,4,4,4,3,4,4,3,5);
  var
    G, A: Array[1..9] of Byte;
    Bog:  Array[1..3] of Byte;
    Hand, RHand, I, B, ParTot:    Byte;
    GTot, ATot, Sing, Doub, Trip: Byte;
    Diff:                         Integer;
    Adjusted:                     Boolean;

begin
  ParTot := 0;  GTot := 0;  ATot := 0;
  Sing   := 0;  Doub := 0;  Trip := 0;
  Write ('Enter handicap: ');  Readln (Hand);
  Write ('Enter gross scores: ');
  Readln (G[1],G[2],G[3],G[4],G[5],G[6],G[7],G[8],G[9]);
  Write ('HOLE #:');
  for I := 1 to 9 do Write (I: 4);
  Writeln;
  Write ('PAR:   ');
```

```
  for I := 1 to 9 do begin
    Write (Par[I]: 4);
    ParTot := ParTot + Par[I];
  end;
  Writeln;
  Write ('GROSS: ');
  for I := 1 to 9 do begin
    Write (G[I]: 4);
    GTot := GTot + G[I];
  end;
  Writeln;
  Write ('ADJUST:');

  { -- Determine # of tripple and double bogeys allowed. }
  if Hand > 9 then begin
    Bog[3] := Hand - 9;
    Bog[2] := 9 - Bog[3];
    Bog[1] := 0
    end
  else begin
    Bog[3] := 0;
    Bog[2] := Hand;
    Bog[1] := 9 - Bog[2];
  end;

  { -- Adjust the gross scores by Handicap. }
  for I := 1 to 9 do begin
    Diff := G[I] - Par[I];
    Adjusted := False;
    B := 3;
    while not Adjusted and (B > 0) do begin
      if (Bog[B] > 0) and (Diff >= B) then begin
        A[I] := Par[I] + B;
        Dec(Bog[B]);
        Adjusted := True;
      end;
      Dec(B);
    end;
    if not Adjusted then
      A[I] := G[I];
  end;  { -- for I }

  { -- Display the adjusted scores and totals. }
  for I := 1 to 9 do begin
    Write (A[I]: 4);
    ATot := ATot + A[I];
  end;
  Writeln;  Writeln;
  Writeln ('PAR TOTAL: ', ParTot);
  Writeln ('GROSS TOTAL: ', GTot);
  Writeln ('ADJUST TOTAL: ', ATot);
  Writeln ('ROUND HANDICAP: ', ATot - ParTot);
end.
```

```
{3.1}
program Thr1T92;
{ -- This program will move a triangle of GTEDS around screen. }
uses Crt;
  const
    A: Array[1..7] of String[11] =
       ('           ',
        '     G     ',
        '    T T    ',
        '   E   E   ',
        '  D     D  ',
        ' SDETGTEDS ',
        '           ');
   var
     Row, Col, I: Integer;
     Ch: Char;

begin
  ClrScr;
  Row := 9;  Col := 34;
  repeat
    for I := 1 to 7 do begin
      GotoXY (Col, Row + I);
      Writeln (A[I]);
    end;
    for I := 1 to 1000 do
      if KeyPressed then Ch := ReadKey;
    case Upcase(CH) of
      'I' : Dec(Row);
      'M' : Inc(Row);
      'J' : Dec(Col);
      'K' : Inc(Col);
    end;
    if Row = 0 then begin
      Row := 1;  Ch := ' ';  end;
    if Col = 0 then begin
      Col := 1;  Ch := ' ';  end;
    if Row = 18 then begin
      Row := 17; Ch := ' ';  end;
    if Col = 69 then begin
      Col := 68; Ch := ' ';  end;
  until Ch = Chr(27);
  Ch := ' ';
end.
```

```
{3.2}
program Thr2T92;
{ -- This program will display a date in 1992 after # of days. }
  const
    Day:   Array[1..6] of String[10] = ('TUESDAY', 'WEDNESDAY',
           'THURSDAY', 'FRIDAY', 'SATURDAY', 'MONDAY');
    Month: Array[1..12] of Integer =
           (31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    MName: Array[1..12] of String[10] = ('JANUARY', 'FEBRUARY',
           'MARCH', 'APRIL', 'MAY', 'JUNE', 'JULY', 'AUGUST',
           'SEPTEMBER', 'OCTOBER', 'NOVEMBER', 'DECEMBER');
  var
    X, D, I, Sum: Integer;

begin
  Write ('Enter X: ');  Readln (X);
  Inc(X);
  D := (X MOD 6) + 1;
  Write(Day[D], ' ');
  X := X + (X + 1) div 6;
  Sum := 0;  I := 1;
  while Sum + Month[I] < X do begin
    Sum := Sum + Month[I];  Inc(I);
  end;
  Writeln (MName[I], ' ', X - Sum);
  if Day[D] = 'SATURDAY' then begin
    Inc(X);
    while Sum + Month[I] < X do begin
      Sum := Sum + Month[I];  Inc(I);
    end;
    Writeln ('SUNDAY ', MName[I], ' ', X - Sum);
  end;
end.



{3.3}
program Thr3T92;
{ -- This program will release program modules for PWS. }
  var
    Name, Prog:   Array [1..9] of String[10];
    Comp, Rel:    Array [1..9] of String[1];
    St, Module:   String[20];
    I, J, L, Num: Byte;
    AllDone, ModComp, ModRel: Boolean;

begin
  I := 0;  Num := 0;  AllDone := False;  ModRel := False;
  repeat
    I := Num + 1;
    Write ('Enter name, program: ');  Readln (St);
      L := Length(St);
      Name[I] := Copy(St, 1, L - 5);
      Prog[I] := Copy(St, L - 3, 4);
```

```
      { -- Find previous Name/Prog or make addition. }
      J := 1;
      while (J < I) and ((Name[J] <> Name[I]) or
      (Prog[J] <> Prog[I])) do
        Inc(J);
      I := J;
      if I > Num then Num := I;

    Write ('Enter completed, release: ');  Readln (St);
      Comp[I] := Copy(St, 1, 1);
      Rel[I]  := Copy(St, 3, 1);
      if Rel[I] = 'Y' then Comp[I] := 'Y';
      ModComp := (Comp[I] = 'Y');

    { -- Check if Module completed by all at least 1 released. }
    if ModComp then begin
      ModRel := False;
      for J := 1 to Num do
        if (Prog[J] = Prog[I]) then begin
          if (Comp[J] <> 'Y') then
            ModComp := False;
          if (Rel[J] = 'Y') then
            ModRel := True;
        end;

      { -- If Module completed by all and 1 or more released. }
      if ModComp and ModRel then begin
        Writeln ('MODULE ', Prog[I], ' HAS BEEN RELEASED');
        Module := Prog[I];
        for J := 1 to Num do
          if Prog[J] = Module then Prog[J] := '';
        AllDone := True;
        for J := 1 to Num do
          if Prog[J] <> '' then AllDone := False;
      end;
    end;  { -- If ModComp }
  until AllDone;
end.
```

```
{3.4}
program Thr4T92;
{ -- This program will produce acronyms for phone numbers. }
  const
    B: Array [1..18] of String[5] = ('AGENT', 'SOAP', 'MONEY',
    'JEWEL', 'BALL', 'LOANS', 'CARE', 'SAVE', 'CALL', 'PAVE',
    'KEEP', 'KINGS', 'KNIFE', 'KNOCK', 'JOINT', 'JUICE',
    'LOBBY', 'RATE');
    L1: String[9] = ' ADGJMPTW';
    L2: String[9] = ' BEHKNRUX';
    L3: String[9] = ' CFILOSVY';
  var
    I, J, K, L: Byte;
    Ph, Num:    String[8];
    P4, P5, X:  String[5];
    C:          String[1];
    A:          Array[1..18] of String[5];

begin
  { -- Sort the data alphabetically. }
  for I := 1 to 18 do A[I] := B[I];
  for I := 1 to 17 do
    for J := I + 1 to 18 do
      if A[I] > A[J] then begin
        X := A[I];  A[I] := A[J];  A[J] := X;
      end;

  Write ('Enter phone #: ');  Readln (Ph);
  P4 := Copy(Ph, 5, 4);  P5 := Copy(Ph, 3, 1) + P4;
  { -- Convert words to number strings }
  for I := 1 to 18 do begin
    L := Length(A[I]);  Num := '';
    for J := 1 to L do begin
      K := 2;  C := Copy(A[I], J, 1);;
      while (L1[K] <> C) and (L2[K] <> C) and (L3[K] <> C) do
        Inc(K);
      Num := Num + Chr(48 + K);
    end;
    if (L = 4) and (Num = P4) then
      Writeln (Copy(Ph, 1, 4), A[I])
    else if (L=5) and (Num = P5) then begin
      Write (Copy(Ph, 1, 2),  Copy(A[I], 1, 1), '-');
      Writeln (Copy(A[I], L - 3, 4));
    end;
  end;
end.
```

```
{3.5}
program Thr5T92;
{ -- This program will find seven 7-digit squares in base 8. }
  var
    Num1V, Num2, Power, Num: LongInt;
    I, J, K, X, Digit, SNum: Integer;
    Num1, NumSt:             String[4];
    Square:                  String[7];
    Dup:                     Array[0..7] of Boolean;
    Valid:                   Boolean;

begin
  Num := 1242;  SNum := 0;
  repeat
    Str (Num, Num1);

    { -- Convert Num1 to base 10 number Num1V }
    Num1V := 0;
    for I := 1 to 4 do begin
      Digit := Ord(Num1[I]) - Ord('0');
      Power := 1;
      for J := 1 to Length(Num1) - I do
        Power := Power * 8;
      Num1V := Num1V + Digit * Power;
    end;

    Num1V := Num1V * Num1V;

    Square := '';  Valid := True;
    for I := 0 to 7 do
      Dup[I] := False;

    { -- Convert Num1V to Base8 number }
    J := Trunc(Ln(Num1V) / Ln(8));
    repeat
      Power := 1;
      for K := 1 to J do  Power := Power * 8;
      X := Trunc(Num1V / Power);
      { -- Check for duplicate digits. }
      if not Dup[X] then begin
        Dup[X] := True;
        Square := Square + Chr(48 + X);
        Num1V := Num1V - X * Power;
        end
      else
        Valid := False;
      Dec(J);
    until (J < 0) or not Valid;

    if Valid then begin
      Inc(SNum);
      Writeln (Square, '  ', Num);
    end;
```

```
    { -- increment to next base 8 number }
    repeat
      Inc(Num);
      Str(Num, NumSt)
    until (Pos('8',NumSt) = 0) and (Pos('9',NumSt) = 0);

  until SNum = 7;
end.



{3.6}
program Thr6T92;
{ -- This program will find 3 distinct integers that are pairwise
  -- relatively prime such that they sum to N. }
var
  X, Y, Z, N, I: Integer;
  Found:         Boolean;

begin
  Write ('Enter N: ');  Readln (N);
  X := 2 + (N mod 2);  Found := False;
  while (X < N div 3) and not Found do begin
    Y := X + 1;
    while (Y < (N - X) div 2) and not Found do begin
      Z := N - X - Y;  Found := True;
      for I := 2 to Y do
        if ((X mod I = 0) and (Y mod I = 0)) or
           ((X mod I = 0) and (Z mod I = 0)) or
           ((Y mod I = 0) and (Z mod I = 0)) then
          Found := False;
      if Found then
        Writeln (X, ' + ', Y, ' + ', Z, ' = ', N)
      else
        Inc(Y);
    end;
    Inc(Z);
  end;
end.
```

```
{3.7}
program Thr7T92;
{ -- This program will print combinations of 6 soccer players. }
uses Crt;
  var
    A:    Array [1..9] of Integer;
    Name: Array [1..9] of String[10];
    X:    String[10];
    Ch:   Char;
    I, J, M, L, N, S, Sub: Byte;

begin
  Name[1] := 'ANDY';  Name[2] := 'DAN';  Name[3] := 'DOUG';
  Name[4] := 'JACK';  Name[5] := 'MIKE'; Name[6] := 'YEHIA';

  Write ('Enter number of substitutes: ');  Readln (Sub);
  L := 6 + Sub;
  for I := 7 to L do begin
    Write ('Enter name: ');  Readln (Name[I]);
  end;
  { -- Sort names with substitutes. }
  for I := 1 to L - 1 do
    for J := I + 1 to L do
      if Name[I] > Name[J] then begin
        X := Name[I];  Name[I] := Name[J];  Name[J] := X;
      end;

  M := 6;
  for I := 1 to M do   A[I] := M - I + 1;
  N := 1;  A[1] := A[1] - 1;  S := 0;
  while N <= M do begin
    A[N] := A[N] + 1;
    if N > 1 then
      for I := N-1 downto 1 do A[I] := A[I+1] + 1;
    if A[N] <= L - N + 1 then begin
      Inc(S);
      Write (S, ' ',Name[A[M]]);
      for I := M - 1 downto 1 do Write (',',Name[A[I]]);
      Writeln;
      N := 0;
      if S mod 24 = 0 then Ch := ReadKey;
    end;
    Inc(N);
  end;
end.
```

```
{3.8}
program Thr8T92;
{ -- This program displays the Bill Date and the Due Date. }
{ -- January 1, 1992 was a Wednesday. }
  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 Integer =
           (31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    Dname: Array [1..7] of String[9] = ('TUESDAY', 'WEDNESDAY',
           'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY', 'MONDAY');
  var
    I, T, H, Hol, Wkend, X, MNum, Cycle, NumDays: Integer;
    Mhol, Dhol: Array [1..12] of Integer;
    Day, Days:  Array [1..2]  of Integer;

begin
  Write ('Enter month of bill: ');  Readln (MNum);
  Write ('Enter cycle number: ');   Readln (Cycle);
  Write ('Enter number of days: '); Readln (NumDays);
  H := 1;
  Write ('Enter holiday MM, DD: ');  Readln (Mhol[H], Dhol[H]);
  while Mhol[H] > 0 do begin
    H := H + 1;
    Write ('Enter holiday MM, DD: ');  Readln (Mhol[H], Dhol[H]);
  end;
  Dec(H);  Writeln;
  Days[1] := 0;
  for I := 1 to MNum - 1 do
    Days[1] := Days[1] + Mon[I];
  Day[1] := 3 * Cycle - 2;  Day[2] := Day[1] + NumDays;
  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;
      if Day[T] > Mon[MNum] then begin
        Day[T] := Day[T] - Mon[MNum];
        Inc(MNum)
      end;
      for I := 1 to H do
        if (Mhol[I] = MNum) and (Dhol[I] = Day[T]) then begin
          Inc(Day[T]);
          Inc(Days[T]);  Hol := 1;
        end;
      X := Days[T] mod 7;
      if (X = 4) or (X = 5) then begin  { -- Sat. or Sun. }
        Inc(Day[T]);
        Inc(Days[T]);  Wkend := 1;
      end;
    end;  { -- while }
    if T = 1 then Write ('BILL ') else Write ('DUE ');
```

```
    Write   ('DATE: ', Dname[X+1], ' ', Mname[MNum], ' ');
    Writeln (Day[T]);
  end;  { -- for T }
end.



{3.9}
program Thr9T92;
{ -- This program will calculate the area of a polygon room. }
  var
    I, L, Sides, Code, Sum, Area: Integer;
    Mov:  String[3];
    Dir:  Array[1..10] of String[1];
    Dist: Array[1..10] of Integer;

begin
  Write ('Enter number of sides: ');  Readln (Sides);
  for I := 1 to Sides do begin
    Write ('Enter movement: ');  Readln (Mov);
    Dir[I] := Copy(Mov, 1, 1);
    L := Length(Mov);
    Mov := Copy(Mov, 2, L - 1);
    Val(Mov, Dist[I], Code);
    { -- Subtract Down and Left directions }
    if (Dir[I] = 'D') or (Dir[I] = 'L') then
      Dist[I] := -Dist[I];
  end;
  { -- Multiply length by width to obtain rectangle area, }
  { -- then add or subtract area from overall area. }
  I := 1;  Sum := 0;  Area := 0;
  repeat
    Sum := Sum + Dist[I];
    Area := Area + (Sum * Dist[I+1]);
    Inc(I, 2);
  until (I > Sides);
  Writeln ('AREA = ', Abs(Area), ' SQUARE FEET');
end.
```

```
{3.10}
program Thr10T92;
{ -- This program will display the reasons a Rubik's Cube is }
{ -- unsolvable. }
  const
   Side:  Array[1..6] of String[7] =
          ('TOP:   ', 'FRONT: ', 'RIGHT: ', 'BACK:  ',
           'LEFT:  ', 'BOTTOM:');
   EdgeS: String[60] =
   'T2P2 T6R2 T8F2 T4L2 F4L6 F6R4 R6P4 P6L4 F8B2 R8B6 P8B8 L8B4';
  var
    Col:       Array[1..6, 1..9] of String[1];
    I, J, K:   Byte;
    MidUnique: Boolean;
    Colors:    String[17];
    S1, S2, N1, N2, ENum: Byte;


begin
  for I := 1 to 6 do begin
    Write ('Enter colors on ', Side[I], ' ');  Readln (Colors);
    for J := 1 to 9 do
      Col[I,J] := Copy(Colors, J * 2 - 1, 1);
  end;

   MidUnique := True;
   for I := 1 to 5 do
     for J := I + 1 to 6 do
       if Col[I, 5] = Col[J, 5] then
         MidUnique := False;

   if not MidUnique then
     Writeln ('COLORS ON MIDDLE SQUARES ARE NOT UNIQUE');

   ENum := 0;
   for K := 1 to 12 do begin
     S1 := Pos(EdgeS[K*5 - 4], 'TFRPLB');
     N1 := Ord(EdgeS[K*5 - 3]) - 48;
     S2 := Pos(EdgeS[K*5 - 2], 'TFRPLB');
     N2 := Ord(EdgeS[K*5 - 1]) - 48;
     if Col[S1, N1] = Col[S2, N2] then
       Inc(ENum);
   end;

   Writeln ('NUMBER OF EDGE PIECES HAVING SAME COLOR: ', ENum);
end.
```