

ملحق (1)

برامج بعض التمارين

obbeikandl.com

obeikandi.com

برامج تمارين الباب الثالث

تمرين 3-9

رقم 1

```
PROGRAM fromCtoF;
VAR c: INEGER;
    f: REAL;
BEGIN
    c:=1;
    WHILE (c<=100) DO
    BEGIN
        f := 9.*c/5 + 32;
        WRITELN(c, f:5:2);
        c := c+1;
    END;
END.
```

```
PROGRAM fromCtoFusingREPEAT;
VAR c: INTEGER;
    f : REAL;
BEGIN
    c:=20;
    REPEAT
        f := 9.*c/5 +32;
        WRITELN(c, ' ', f:10:2);
        c := c+1;
    UNTIL
        c>40
END.
```

```

PROGRAM fromCtoFusingREPEAT;
VAR c: INTEGER;
    f : REAL;
BEGIN
FOR c:=20 TO 40 DO
BEGIN
    f := 9.*c/5 +32;
    WRITELN(c, ' ', f:10:2);
END;
END.

```

تمرین رقم 3

```

PROGRAM findMin;
VAR xmin , x, k : INTEGER ;
BEGIN
    xmin := 100;
    K:=0;
WHILE K<5 DO
BEGIN
    WRITE
        (' ENTER A VALUE--');
    READ(x);
    IF x < xmin THEN
        xmin:=x;
    K := K+1;
END;
WRITELN( 'MINIMUM=', xmin);
END.

```

```

PROGRAM findMin;
VAR xmin , x, k : INTEGER ;
BEGIN
    xmin := 100;
    K:=0;
    REPEAT
        WRITE
            (' ENTER A VALUE--');
        READ(x);
        IF x < xmin THEN
            xmin:=x;
        K := K+1;
    UNTIL
        K = 5;
    WRITELN( 'MINIMUM=',xmin);
END.

```

```

PROGRAM findMin;
VAR xmin , x, k : INTEGER ;
BEGIN
    xmin := 100;
    FOR k := 1 TO 5 DO
        BEGIN
            WRITE
                (' ENTER A VALUE--');
            READ(x);
            IF x < xmin THEN
                xmin:=x;
        END;
    WRITELN( 'MINIMUM=',xmin);
END.

```

```

program ave;
VAR av1, av2, sum1, sum2, g : REAL;
    n1, n2 , s: INTEGER;
BEGIN
    SUM1:=0; n1:=0; sum2:=0;
    n2:=0;
    WRITE('enter 1 for male',
        ' 2 for female-->');
    READ(s);
    WRITE('enter grade-->');
    READ(g);
    WHILE g>=0 DO
    BEGIN
        IF s=1 THEN
            BEGIN
                sum1 := sum1 + g;
                n1 := n1+ 1;
            END
        ELSE
            BEGIN
                sum2 := sum2 + g;
                n2 := n2 +1;
            END;
        WRITE('ENTER 1 FOR MALE',
            ' 2 FOR FEMALE-->');
        READ(s);
        WRITE('ENTER GRADE-->');
        READ(g);
    END;
    av1 := sum1/n1;
    av2 := sum2/n2;
    Writeln('AVERAGE OF MALE
    STUDENTS=', av1:6:2);
    Writeln('AVERAGE OF FEMALE
    STUDENTS=', av2:6:2);
    END.

```

```
program ave;
VAR av1, av2, sum1, sum2, g : REAL;
    n1, n2 , s: INTEGER;
BEGIN
    SUM1:=0; n1:=0; sum2:=0; n2:=0;
    WRITE('enter 1 for male, 2 for
female-->');
    READ(s);
    WRITE('enter grade-->');
    READ(g);
    REPEAT
        IF s=1 THEN
            BEGIN
                sum1 := sum1 + g;
                n1 := n1+ 1;
            END
        ELSE
            BEGIN
                sum2 := sum2 + g;
                n2 := n2 +1;
            END;
        WRITE('ENTER 1 FOR MALE, 2 FOR
FEMALE-->');
        READ(s);
        WRITE('ENTER GRADE-->');
        READ(g);
        UNTIL
            g < 0 ;

    av1 := sum1/n1;
    av2 := sum2/n2;
    WRITELN('AVERAGE OF MALE
STUDENTS=', av1:6:2);
    WRITELN('AVERAGE OF FEMALE
STUDENTS=', av2:6:2);
END.
```

```

Program ave;
LABEL out;
VAR av1, av2, sum1, sum2, g : REAL;
    K, n1, n2, s : INTEGER;
BEGIN
    SUM1 := sum2 := 0; n2 := 0;
WRITE ( ' enter 1 for male, 2 for female >' );
    READ (s) ;
WRITE ( ' enter grade - - >' );
    READ (g) ;
    FOR K: =1 to 100 Do
        BEGIN
            IF g<0 THEN GOTO out;
            IF s=1 THEN
                BEGIN
                    Sum1 := sum1 + g;
                    n1 : n1 + 1 ;
                END
            ELSE
                BEGIN
                    Sum2 := sum2 + g ;
                    n2 := n2 + 1 ;
                END ;
        END ;
WRITE ( ' ENTER 1 FOR MALE, 2 FOR FEMALE → ' ) ;
    READ (s) ;
WRITE ( ' ENTER GRADE → ' ) ;
    READ (g) ;
    END; {for}

    Out : av1 := sum1 / n1 ;
        Av2 := sum2 / n2 ;
WRITELN ( ' AVERAGE OF MALE STUDENTS = ', av1
        : 6 : 2 ) ;
WRITELN ( ' AVERAGE OF FEMALE STUDENTS = ',
        av2 : 6 : 2 ) ;

END.

```

```

PROGRAM grades;
  VAR g : REAL
      C: CHAR;
  BEGIN
    WRITE ( ' Enter grade - - > ' );
    READ (g) ;
    IF ( g<50 ) THEN c:= 'F' ;
    IF ( g>50 ) AND ( g < 70 ) THEN
      C: = ' p ' ;
    IF ( (g>70) AND ( g<85) ) THEN
      C: = ' G ' ;
    IF ( g>= 85 ) THEN c:= ' A ' ;
    WRITE ( ' Your grade is ', c ) ;
  END.

```

7-ج

```

PROGRAM grades;
VAR g : REAL;
    c : CHAR;
    k : INTEGER;
BEGIN
  WRITE('Enter grade-->');
  READ(g);
  REPEAT
    IF( g<50 ) THEN c:='F';
    IF( (g>=50) AND (g < 70) ) THEN
c:='P';
    IF( (g>=70) AND (g<85)) THEN c:='G';
    IF( g>=85 ) THEN c:='A';
    WRITELN('Your grade is ',c);
    WRITE('Enter grade-->');
    READ(g);
  UNTIL
    g<0
  END.

```

تمرین رقم 9

```
PROGRAM stars;  
VAR k,i: INTEGER;  
BEGIN  
  FOR k:=1 TO 4 DO  
    BEGIN  
      FOR i:= 1 TO k*k DO  
        WRITE('*');  
      WRITELN;  
    END;  
  END.  
END.
```

برامج تمارين الباب الرابع

تمارين 4-9

رقم 5

```
PROGRAM daysInMonth;
TYPE month=(jan, feb, mar, apr,
may, jun, jul, aug, sep, oct, nov, dec);
VAR m : months;
    days:ARRAY[month]OF INTEGER;
name : ARRAY[month] OF STRING[10];
BEGIN
    name[jan]:= 'January' ;
days[jan]:=31;
    name[feb]:= 'February' ;
days[feb]:=29;
    name[mar]:= 'March' ;
days[mar]:=31;
FOR m:=jan TO mar DO
    WRITLN('number of days in
',name[m], ' is ',days[m]);
END.
```

```
PROGRAM sort;
VAR gr:ARRAY[1..100]OF REAL;
    first, temp: REAL;
    n, i, k, p : INTEGER;
BEGIN
    WRITE('Enter number of items-->');
    READ(n);
    FOR i:=1 TO n DO
        BEGIN
            WRITE('Enter grade',i,'-->');
            READ(gr[i]);
        END;
    FOR i:=1 TO n-1 DO
        BEGIN
            first:=gr[i];
            p:=i;
            FOR k:=i+1 to n DO
                IF gr[k] > first THEN
                    BEGIN
                        first:= gr[k];
                        p:=k;
                    END ;
            temp:=gr[i]; gr[i]:=gr[p];
            gr[p]:=temp;
        END;
    FOR i:=1 TO n DO
        WRITELN(gr[i]:5:2);
    END.
```

```

PROGRAM words;
VAR i,j,k,L:INTEGER;
    c:ARRAY[1..4]OF CHAR;
BEGIN
    c[1]:='O'; c[2]:='M'; c[3]:='A';
    c[4]:='R';
    FOR i:=1 TO 4 DO
        FOR j:=1 TO 4 DO
            FOR k:=1 TO 4 DO
                FOR L:=1 TO 4 DO
                    WRITE(' ',c[i],c[j],c[k],c[L]);
                END.
            END.
        END.
    END.

```

```

PROGRAM payrollFile;
TYPE empRec=RECORD
    num: INTEGER;
    name : STRING[15];
    bas, allow, deduct : REAL;
END;
VAR    k, n : INTEGER;
BEGIN
    Writeln;
    WRITE(' PAYROLL PROGRAM');
    Writeln; Writeln;
    WRITE
        ('Enter number of employees-->');
    READ(n);
    FOR k:=1 TO n DO
    BEGIN
        WRITE('Enter number-->');
        READLN(emRec.num);
        WRITE('Enter Name-->');
        READLN(emRec.name);
    END.

```

```

WRITE('Enter basic salary-->');
  READ(empRec.bas);
  WRITE('Enter allow[k]-->');
  READ(empRec.allow);
  WRITE('Enter deduction-->');
  READ(empRec.deduct);
  WRITE(empFile,empRec) ;
END;
END.

```

رقم 12

```

PROGRAM payrollFile;
TYPE empRec=RECORD
  num: INTEGER;
  name : STRING[15];
  bas, allow, deduct : REAL;
END;
VAR   k, n : INTEGER;
      empFile : FILE OF empRec;
      er : empRec;

BEGIN
  ASSIGN(empFile, 'empFile.dta');
  REWRITE(empFile);
  WRITELN;
  WRITE(' PAYROLL PROGRAM');
  WRITELN; WRITELN;
  WRITE('number of employees?->');
  READ(n);
  FOR k:=1 TO n DO
  BEGIN
    WRITE('Enter number-->');
    READLN(er.num);
    WRITE('Enter Name-->');
    READLN(er.name);
    WRITE('basic salary-->');
    READ(er.bas);
    WRITE('Enter allow[k]-->');
    READ(er.allow);
    WRITE('Enter deduction-->');

```

```

        READ(er.deduct);
    WRITE(empFile,er) ;
END;
END.

```

رقم 13

```

PROGRAM payroll;
TYPE empRec=RECORD
    num: INTEGER;
    name : STRING[15];
    bas, allow, deduct : REAL;
END;
VAR
    k, n : INTEGER;
total : REAL;
empFile : FILE OF empRec;
er : empRec;
BEGIN
    ASSIGN(empFile,'empFile.dta');
    RESET(empFile);
    WRITELN;
    WRITELN('number':10,'Name':15,
'Basic ':10,'Allowance':10,'Deduct ':10,
'Total':10);
    WHILE NOT EOF(empFile) DO
        BEGIN
            READ(empFile,er);
            total:=er.bas+er.allow-
er.deduct;
            WRITELN;
            WRITELN(er.num:10,er.name:15,
er.bas:10:3,er.allow:10:3,
er.deduct:10:3, total:10:3);
        END;
    END.

```

```

PROGRAM schedule;
TYPE week=1..7;
VAR p , p1, p2, p3, p4, p5 :
    SET OF week;
    k: week;
BEGIN
    p1:=[1,3,7];
    p2:=[2,3,6];
    p3:=[1,2,3];
    p4:=[1,3,5];
    p5:=[2,3,5];
    p:=p1*p2*p3*p4*p5;
    IF p=[] THEN
        WRITELN
        (' INTERSECTION is empty.',
         ' No day is common.')
    ELSE
        FOR k:= 1 TO 7 DO
            IF k IN p THEN
                WRITELN(' day number ',k,
                 ' is common');
        END.

```

```

program depart;
var L : array[1..20] of integer;
    boss : array[1..5] of integer;
    emp : array [1..5, 1..20] of integer;
    k, m : integer;
begin
    for k:=1 to 20 do
    begin
        writeln;
        write
        ('who is the boss of emp ',k,'? ');
        read(boss[k]);
    end;
    for m:=1 to 5 do
    begin
        L[m]:=0;
        for k:= 1 to 20 do
            if boss[k]=m then
                begin
                    L[m]:=L[m]+1;
                    emp [m,L [m] ]:=k;
                end;
        end;
        for m:=1 to 5 do
        begin
            writeln;
            write
            ('Employees under boss number ',m,'
are: ');
                for k:= 1 to L[m] do
                    write( emp [m,k] :3);
                end;
        end.
    end.

```

obeikandi.com

برامج تمارين الباب الخامس

تمارين 5-11

رقم (1)

```
PROGRAM list;
USES CRT;
PROCEDURE menu;
BEGIN
    CLRSCR;
    GOTOXY(30,7);
    WRITE('1. List OF Students. ');
    GOTOXY(30,8);
    WRITE('2. List Of Courses. ');
    GOTOXY(30,9);
    WRITE('3. Exit ');
END;

BEGIN
    menu;
END.
```

```

PROGRAM readWrite;
USES CRT;
VAR      x: ARRAY[1..50] OF REAL;
         n , k,L :INTEGER;
PROCEDURE readArray;
BEGIN
    FOR k:=1 TO n DO
    BEGIN
        L:=k+5;
        GOTOXY(30,L);
        WRITE('Enter value of x[',
              k ,']-->');
        READ(x[k]);
    END;
END;
PROCEDURE printArray;
BEGIN
    FOR k:=1 TO n DO
    BEGIN
        L:=k+5;
        GOTOXY(60,L);
        WRITE(x[k]:12:3);
    END;
END;

BEGIN
    CLRSCR;
    GOTOXY(30,5);
    WRITE
    ('HOW many elements do you have? ');
    READ(n);
    ReadArray;
    PrintArray;
END.

```

```
PROGRAM DrawTriangle;
USES CRT;
VAR k: INTEGER;

PROCEDURE printx(n:INTEGER);
VAR k: INTEGER;
BEGIN
    FOR k:=1 TO n DO
        WRITE('x');
        WRITELN;
    END;
BEGIN {main }
    CLRSCR;
    FOR k:= 5 DOWNT0 1 DO
        printx(k);
END.
```

تمرین (7)

```
PROGRAM Compute;
VAR x:REAL; k:INTEGER;
FUNCTION p(x:REAL):REAL;
BEGIN
    p:=5*x*x+2*x-1/x;
END;
BEGIN
    FOR k:=2 TO 10 DO
        BEGIN
            x:=k*0.5;
            WRITELN(x:3:1, p(x):10:3);
        END;
    END.
END.
```

```
PROGRAM compare;
USES CRT;
VAR x, y : REAL;

FUNCTION test(x,y:REAL):BOOLEAN;
BEGIN
    IF x=y THEN test:=TRUE
    ELSE
        test:=FALSE;
END;
BEGIN {main}
    CLRSCR;
    WRITE(' ENTER 2 VALUES-->');
    READ(x,y);
    WRITE(test(x,y));
END.
```

```
PROGRAM computeLog;
USES CRT;
VAR x:REAL;
FUNCTION log(x:REAL):REAL;
VAR k:INTEGER;
    p, sum :REAL;
BEGIN
    k:=1;
    p:=x;
    sum:=x;
    REPEAT
        p:=-p*x;
        k:=k+1;
        sum:=sum+p/k;
    UNTIL
        abs(p/k) <= 1.0e-4;
    log:=sum;
END;
```

```

BEGIN {main}
  clrscr;
  WRITE('Enter x-->');
  READ(x);
  WRITELN('The natural log of ',
          x:5:2,
          ' as computed by this program is '
          ,log(x-1));
  WRITELN('The natural logarithm as
          computed by ln(x) is ',
          ln(x));
END.

```

تمرین (13)

```

PROGRAM TwoEq;
VAR a1,a2,b1,b2,c1,c2,x,y:REAL;

PROCEDURE solve(a1,b1,c1,a2,b2,c2:REAL;
               VAR x,y:REAL);
VAR t:REAL;
BEGIN
  t:=a2/a1;
  c2:=c2-t*c1;
  b2:=b2-t*b1;
  y:=c2/b2;
  x:=(c1-b1*y)/a1;
END;

BEGIN
  WRITE('Enter a1, b1, c1-->');
  READ(a1,b1,c1);
  WRITE('Enter a2, b2, c2-->');
  READ(a2, b2,c2);
  solve(a1,b1,c1,a2,b2,c2,x,y);
  WRITELN(' The solution is x='
          ,x:10:3, ' y=' ,y:10:3);
END.

```

```
PROGRAM recursive;  
  
FUNCTION f(n:INTEGER):REAL;  
VAR sign:INTEGER;  
BEGIN  
    IF (n=1) THEN  
        BEGIN  
            f:=1;  
        END  
    ELSE  
        BEGIN  
            IF ODD(n) THEN sign:=1  
            ELSE sign:=-1;  
            f:=f(n-1)+sign*1.0/n;  
        END;  
    END;  
END;  
BEGIN  
    WRITELN(f(5));  
END.
```

```
PROGRAM makeWords;
VAR c:ARRAY[1..4] OF CHAR;
    n, i :INTEGER;
    p:ARRAY[1..4]OF 1..4 ;
PROCEDURE printWord;
BEGIN
    FOR i:=1 TO n DO
        WRITE(c[p[i]]);
        WRITE(' ');
    END;
PROCEDURE recursive(k:INTEGER);
VAR t:INTEGER;
BEGIN
    FOR t:=1 TO n DO
        BEGIN
            p[k]:=t;
            IF k=n THEN printWord
            ELSE
                recursive(k+1);
        END;
    END;
BEGIN
    c[1]:='O'; c[2]:='M'; c[3]:='A';
    c[4]:='R';
    WRITELN;
    n:=4;
    recursive(1);
END.
```

برامج تمارين الباب السادس

تمارين (3)

```
PROGRAM EmployeeFile;
TYPE
  family = RECORD
    wife : STRING[20];
    children : INTEGER;
  END;
  emp=RECORD
    num : INTEGER;
    name : STRING[20];
    status : CHAR;
    CASE st: CHAR OF
      'S' : ();
      'M' : (fam : family)
    END; {emp }
VAR
  f : FILE OF emp;
  employee : emp;
  fam : family;
  i,nr : INTEGER;

BEGIN {main }
  ASSIGN(f,'VarEmp.dta');
  { Create File }
  REWRITE(f) ;
  FOR i:=1 TO 5 DO WRITELN;
  WRITE('Enter number of employees-->');
  READ(nr);
  FOR i:=1 To nr DO
  BEGIN
    WITH employee DO
    BEGIN
      WRITE('Enter Emp number-->');
```

```

READLN(num);
WRITE('Enter name-->');
READLN(name);
WRITE('Enter social status:',
      'S=single, M=Married-->');
      READLN(status);
      IF( status='M' ) THEN
        WITH fam DO
          BEGIN
            WRITE('Enter wife or',
                  ' husband name-->');
            READLN(wife);
            WRITE('Enter number of',
                  ' children-->');
            READLN(children);
            END;
          END;
        WRITE(f, employee);
      END;{ for }
    CLOSE(f);
  { Display File }
  RESET(f);
  WHILE NOT EOF(f) DO
    BEGIN
      READ(f, employee);
      WITH employee DO
        BEGIN
          WRITE(num:10,name:20,
                employee.status:3);
          IF status='M' THEN
            WITH fam DO
              WRITE(wife:20,children:3);
              WRITELN;
            END;
          END;
        END;
      END.

```

```
PROGRAM pointers;
TYPE pt=^rec;

    studentRec = RECORD
num:INTEGER;
name : STRING[20];
final : REAL;
END;

    rec=RECORD
f1 : studentRec;
f2 : pt;
END;
VAR p, head : pt;
s : studentRec;
i , count: INTEGER;
f : FILE OF studentRec;
BEGIN
    ASSIGN(f,'student.grs');
{ create file }
    REWRITE(f);
    FOR i:= 1 TO 5 DO
    BEGIN

        WRITELN;
        WRITE('Enter num-->');
        READLN(s.num);
        WRITE('Enter name-->');
        READLN(s.name);
        WRITE('Enter final-->');
        READLN(s.final);
        WRITE(f,s)

    END;
END;
```

```

{ read file }
  RESET(f);
  head := NIL;
  count:=0;
  WHILE NOT EOF(f) DO
  BEGIN
    READ(f,s);
    NEW(p);
    p^.f1 := s;
    p^.f2 := head;
    head:=p;
    count:=count+1;
  END;
{ display file backward }
  p := head;
  FOR i:= 1 TO count DO
  BEGIN
    WRITELN(p^.f1.num:5,
            p^.f1.name:20,
            p^.f1.final:12:2);
    p:=p^.f2;
  END;
END.

```

تمرین (9)

```

PROGRAM pointers;
TYPE pt=^rec;
      rec=RECORD
        f1 : REAL;
        f2 : pt;
      END;
VAR p, head : pt;
    i : INTEGER;
    sum , xbar, sumx : REAL;
BEGIN
  head := NIL;
  sum:=0;
  WRITELN;
  FOR i:=1 TO 10 DO

```

```

BEGIN
  NEW(p);
  WRITE('Enter a value-->');
  READ(p^.f1);
  sum:=sum+p^.f1;
  p^.f2 := head;
  head:=p;
END;
xbar:=sum/10;
WRITELN('xbar= ',xbar:8:2);
p := head;
sumx:=0;
FOR i:= 1 TO 5 DO
  BEGIN
    p:=p^.f2;
    sumx := sumx + sqr(p^.f1 - xbar);
  END;
WRITELN('sum of squares = ',
        sumx:8:2);
END.

```

ملحق (2)

الدوال والإجراءات الجاهزة في

تربو باسكال 6

obeikandi.com

الدوال الجاهزة في تربو باسكال 6

Turbo Pascal 6.0 Functions –

Abs
Addr
ArcTan
Chr
Concat
Copy
Cos
CSeg
DiskFree
DiskSize
DosExitCode
DosVersion
DSeg
EnvCount
EnvStr
Eof
Eoln
Exp
FExpand
FilePos
FileSize
Frac
FSearch
GetBkColor
GetColor
GetDefaultPalette
GetDriverName
GetEnv
GetGraphMode
GetMaxColor
GetMaxMode
GetMaxX

GetModeName
GetPaletteSize
GetPixel
GetX
GetY
GraphErrorMsg
GraphResult
Hi
ImageSize
InstallUserDriver
InstallUserFont
Int
IOResult
KeyPressed
Length
Ln
Lo
MaxAvail
MemAvail
Odd
Ofs
Ord
OvrGetBuf
OvrGetRetry
ParamCount
ParamStr
Pi
Pos
Pred
Ptr
Random
ReadKey
RegisterBGIDriver
RegisterBGIFont
Round
SeekEof
SeekEoln
Seg
Sin

SizeOf
SPtr
Sqr
Sqrt
SSeg
Succ
Swap
TextHeight
TextWidth
Trunc
TypeOf
UpCase
WhereX
WhereY

الإجراءات الجاهزة في ترينوباسكال

Turbo Pascal 6.0 Procedures—

Append
Arc
Assign
AssignCrt
Bar
Bar3D
BlockRead
BlockWrite
ChDir
Circle
ClearDevice
ClearViewPort
Close
CloseGraph
ClrEol
ClrScr
Dec
Delay
Delete
DelLine
DetectGraph
Dispose
DrawPoly
Ellipse
Erase
Exec
Exit
FillChar
FillEllipse
FillPoly
FindFirst
FindNext

FloodFill
Flush
FreeMem
FSplit
GetArcCoords
GetAspectRatio
GetDate
GetCBreak
GetDir
GetFAttr
GetFillPattern
GetFillSettings
GetFTime
GetImage
GetIntVec
GetLineSettings
GetMem
GetModeRange
GetPalette
GetTextSettings
GetTime
GetVerify
GetViewSettings
GoToXY
Halt
HighVideo
Inc
InitGraph
Insert
InsLine
Intr
Keep
Line
LineRel
LineTo
LowVideo
Mark
MkDir
Move
MoveRel

MoveTo
MsDos
New
NormVideo
NoSound
OutText
OutTextXY
OvrClearBuf
OvrInit
OvrInitEMS
OvrSetBuf
PackTime
PieSlice
PutImage
PutPixel
Randomize
Read
ReadLn
Rectangle
Release
Rename
Reset
RestoreCrtMode
Rewrite
Rmdir
RunError
Sector
Seek
SetActivePage
SetAllPalette
SetAspectRatio
SetBkColor
SetCBreak
SetColor
SetDate
SetFAttr
SetFillPattern
SetFillStyle
SetFTime
SetGraphBufSize

SetGraphMode
SetIntVec
SetLineStyle
SetPalette
SetRGBPalette
SetTextBuf
SetTextJustify
SetTextStyle
SetTime
SetUserCharSize
SetVerify
SetViewPort
SetVisualPage
SetWriteMode
Sound
Str
SwapVectors
TextBackground
TextColor
TextMode
Truncate
UnpackTime
Val
Window
Write
WriteLn

ملحق (3)

معجم المصطلحات في لغة باسكال

نقدم في هذا الملحق قائمة بالمصطلحات التي وردت في هذا الكتاب باللغتين الإنجليزية والعربية ، علماً بأن الرقم الموجود بينهما يشير إلى البند الذي ورد فيه هذا المصطلح .

وهنا تجب الملاحظة بأن هذه المصطلحات قد تختلف من كتاب إلى آخر، إذ إنها لم تعتمد بعد ، ولكن اختيارها تم بناءً على اجتهاد شخصي مع استخدام المصطلحات التي أصبحت مألوفة في الرياضيات و المصطلحات التي استخدمها أكثر من كتاب . و لا نرى - على أية حال - أن اختلاف المصطلحات يجب أن يكون عائقاً في عملية التعريب ، فإن العمل الدؤوب في هذا المجال، و توفر المراجع العربية بغزارة في علم الحاسوب ، هو الكفيل في النهاية لإبقاء الجيد من المصطلحات ، واستبدال غير مقبول .

Absolute Value	القيمة المطلقة (5-5)
Actual Parameter	بارامتر فعلي (5-3)
Algorithm	خوارزمية (1-1)
Arctan	دالة معكوس الظل (5-5)
Arithmetic Expression	عبارة حسابية (2-4)
Array	مصفوفة (4-3)
Array Index	دليل المصفوفة (4-3)

Assignment Statement		جملة تعيين (2-6)
Absolute Value		القيمة المطلقة (5-5)
Bit		خانة ثنائية (2-3)
Block		قالب (5-4)
Boolean Expression		عبارة منطقية (2-4)
Call-by-value		استدعاء بالقيمة (5-3)
Call-by-reference		استدعاء بالمرجع (5-6)
Character Type		نوع رمزي (2-3)
Compound Statement		جملة مركبة (2-2)
Constant		ثابت (2-3 ، 6-6)
Data		بيانات (1-5)
Data Processing		معالجة البيانات
Data Structure		تركيبية بيانات (6-7)
Declaration of variable		تحديد المتغيرات (2-5)
Direct Access		تركيبية وصول مباشر (4-3)
Dynamic Data		تركيبية حركية للبيانات (6-7)
EOF		نهاية الملف (4-6)
Eoln		نهاية السطر (5-8)
Enumeration Type		النوع المسرود (4-2)
Field		مجال (6-4)
File		ملف (4-7)

Floating-point	النقطة العائمة (2-3)
Formal Parameter	بارامتر شكلي (5-3)
Function	دالة (5-4)
Fraction	الجزء الكسري (5-5)
Global Variable	متغير عام (5-6)
Graphics	الرسوم (5-7)
Heading	المقدمة (5-4)
Identifier	معرف (2-5)
Indentation	التمائل (2-8)
Integer Type	النوع الصحيح (2-3)
Intersection of Sets	تقاطع الفئات (4-9)
Iterative	تكراري (5-8)
Label	عنوان (6-6)
Linked List	قائمة متصلة (6-8)
List Head	رأس القائمة (6-8)
Loop	الحلقة (5-7)
Main Memory	ذاكرة رئيسية (4-3)
Main Program	البرنامج الرئيسي (5-2)
Nesting	احتواء (5-7)
Nil	لا شيء (6-7)
Operator	مؤثر (2-4)
Ordinal Number	رقم الترتيب (4-2)

Over flow Error	خطأ الفيض (5-7)
Packed Array	مصفوفة مترابطة (4-6)
Parameter List	قائمة البارامترات (5-7)
Pointer	مؤشر (6-7)
Predecessor Function	دالة العنصر السابق (5-7)
Procedure	إجراء (5-3)
Union of Sets	اتحاد الفئات (4-9)
Read Statement	جملة القراءة (2-7)
Real	كسري (حقيقي) (2-3)
Record	سجل (6-3، 4-7)
Recursive	تتابعي (5-8)
Reserved Word	كلمة محجوزة (2-4)
Set	فئة (4-10)
Sorting	الترتيب (4-4)
Standard Function	دالة قياسية (5-5)
Statement	جملة (2-2، 2-6)
Statement Label	عنوان الجملة (3-8)
Static Data Structure	تركيبية بيانات ساكنة (6-7)
String	نصيد (4-6)
Structured Programming	برمجة هيكلية (2-7)
Subrange	مدى جزئي (4-2)
Successor Function	دالة العنصر اللاحق (4-2)

Syntax Diagram	الشكل اللغوي (2-2)
Terminator	مؤشر النهاية (2-2)
Text File	ملف نصي (4-8)
Type Mismatch Error	خطأ عدم تناظر النوع ((6-6))
Type Statement	جملة النوع (6-6)
Value Parameter	بارامتر قيمة (5-6)
Variable Parameter	بارامتر متغير (5-6)
Type Statement	جملة النوع (6-6)
Variant Field Record	سجل ذو مجال متغير (6-3)
Write Statement	جملة الكتابة (2-7)

المراجع

ا- باللغة العربية

- 1- البرمجة بلغة باسكال ، تأليف آرثر م كيلر، ترجمة الدكتور أسامة إبراهيم الدسوقي ، سمير ابراهيم شاهين ، الدار الدولية للنشر والتوزيع (1990) القاهرة .

ب- باللغة الإنجليزية

- 1- William Findley & David Watt, ' Pascal an Introduction to Methodical Programming ' , Pitman Publishing, 1985.
- 2- S. Eisenbach & C. Sadler, ' Pascal For Programmers ' , Spriger-Verlag, 1981.
- 3- V. A. Dyck, J.D. Lawson, J.A. Smith, R.J. Beach, 'Computing, An Introduction To Problem Solving with Pascal', Reston Publishing, 1982.

