

```

imean := imean + i;
r1mean := r1mean + r1;
r2mean := r2mean + r2;
end;
imean := imean / count;
r1mean := r1mean / count;
r2mean := r2mean / count;
writeln;
writeln (imean:10:3, ' ', r1mean:10:3, ' ', r2mean:10:3);
close (f);
end.

```

Si el formato de un número leído de un archivo de texto es incorrecto, el programa produce un error de I/O.

Por ejemplo, leer el número 50,000 con una variable del tipo INTEGER causa error, ya que el número más grande permitido para ese tipo es de 32,767. Similarmente, leer el número 32.1 en una variable de tipo INTEGER causa error por el punto decimal (es ilegal para el tipo Integer). El tipo REAL tiene menos restricciones, ya que estos números pueden ser leídos con o sin punto decimal y en notación normal o científica.

## APENDICE A.-REFERENCIA COMPLETA DE FUNCIONES

La definición de ISO Standar Pascal incluye un número de "funciones estándar" construidas dentro del lenguaje, y no necesitan ser declaradas ni codificadas dentro del programa fuente. Esas funciones caen dentro de dos grupos básicos: 1) Las funciones matemáticas, las cuales proveen operaciones fundamentales, como la raíz cuadrada, el valor absoluto, los logaritmos naturales, las funciones trigonométricas; y 2) las funciones de transferencia, las cuales permiten la relación entre tipos incompatibles, como enteros y de carácter.

Muchos libros refieren los valores pasados a una función estándar como "argumentos". Este término es común dentro del argot matemático, por lo que puede ser confundido con el término parámetro. No hay diferencia mas que de término.

Turbo Pascal implementa todas las funciones del ISO Standar Pascal. A continuación se muestra un glosario completo. También se han añadido las propias del IBM PC.

### LEYENDO NUMEROS DE LOS ARCHIVOS DE TEXTO

Los archivos de texto no solamente pueden almacenar strings o enteros, sino también números. Los números, sin embargo, no se almacenan en el archivo en forma binaria, sino como caracteres. Por ejemplo, en RAM el valor entero 20,545 es almacenado como bytes con el valor hexadecimal 0010 0000 0100 0001. Pero en un archivo de texto, el número 20,545 es almacenado como caracteres "2", "0", "5", "4", "5", cada uno en un byte. Cuando se lee un número de un archivo de texto, Turbo Pascal convierte el string de caracteres a formato binario de enteros.

Asimismo, cuando se lee un número de un archivo de texto, Turbo Pascal convierte los caracteres de espacio en blanco en caracteres de espacio. Entonces, los caracteres siguientes hasta el primer carácter no-espacio se convierten a un número entero. De esta manera, cuando se lee un string que contiene un número, Turbo Pascal combina los caracteres de espacio con el número entero.

```

count := count + 1;
writeln (i:10, ' ', r1:10:3, ' ', r2:10:3);
readln (i, r1, r2);

```

**Chr**  
SINTAXIS: función Chr(i:integer);  
DESCRIPCION: Chr regresa el carácter ASCII que corresponde al número del parámetro.

**Circle [GRAPH UNIT]**  
SINTAXIS: procedure Circle(x:integer; Radius:word);  
DESCRIPCION: Circle dibuja un círculo con el centro en el punto (x,y) y un radio de Radius.

**ClearDevice [GRAPH UNIT]**  
SINTAXIS: procedure ClearDevice;  
DESCRIPCION: ClearDevice borra el contenido de la pantalla.

**ClearViewPort [GRAPH UNIT]**  
SINTAXIS: procedure ClearViewPort(x1,y1,x2,y2:integer);  
DESCRIPCION: ClearViewPort borra el contenido de la pantalla en el área del rectángulo de la pantalla definido por las coordenadas (x1,y1) y (x2,y2).

## ABS

SINTAXIS: function Abs (r:real): real;

function Abs (l:integer): Integer;

DESCRIPCION: Abs regresa el valor absoluto del argumento. El resultado de la funcion es del mismo tipo (real o Integer) como el parámetro.

## Addr

SINTAXIS: function Addr (Var Variable):Pointer;

DESCRIPCION:Addr regresa la dirección de una variable, type, constante o procedimiento. El resultado es un tipo Pointer.

## Append

SINTAXIS: procedure Append (Var F:Text);

DESCRIPCION Append abre un archivo de texto de escritura y posiciona el apuntador de archivo al final de este.

## Arc [GRAPH UNIT]

SINTAXIS: procedure Arc(x,y:integer; StAngle, EndAngle, Radius:word);

DESCRIPCION:Arc dibuja un círculo alrededor de las coordenadas x:y con un radio de Radius.El círculo empieza a dibujarse en sentido de las manecillas del reloj desde StAngle y termina en EndAngle.

## ArcTan

SINTAXIS: function ArcTan(R:real):real;

DESCRIPCION:ArcTan regresa el arctangente del parámetro.

## Assign

SINTAXIS: procedure Assign(Var F:File;Name:string);

DESCRIPCION:Assign une el archivo de la variable F al archivo llamado Name.

## AssignCrt [CRT UNIT]

SINTAXIS: procedure AssignCrt(Var F: Text);

DESCRIPCION:AssignCrt permite al usuario direccionar la salida a la pantalla escribiendo al archivo F.

## Bar [GRAPH UNIT]

SINTAXIS: procedure Bar(x1,y1,x2,y2:integer);

DESCRIPCION:Bar llena el área del rectángulo de la pantalla.

## Bar3D [GRAPH UNIT]

SINTAXIS: procedure Bar3D(x1,y1,x2,y2:integer; Depth:word; Top:boolean);

DESCRIPCION:Bar3D llena en tres dimensiones el área del rectángulo de la pantalla. La dimension de la profundidad se especifica con Depth. Si Top es TRUE, el procedimiento dibuja en tercera dimension en la parte superior del rectángulo.

## BlockRead

SINTAXIS: procedure BlockRead(Var F:File; Var B:Type;

NumRecs:integer;Var RecsRead:integer);

DESCRIPCION:BlockRead intenta leer NumRecs registros del archivo F sin tipo en el buffer B.RecsRead indica el número de los registros ya leídos.Note que el parámetro RecsRead es solo usado en las versiones PC/MS-DOS.

## BlockWrite

SINTAXIS: procedure BlockWrite(Var F:File; Var B:Type; NumRecs:integer);

DESCRIPCION:BlockWrite escribe los NumRecs registros desde el buffer B al archivo sin tipo F.

## ChDir

SINTAXIS: procedure ChDir(S:string);

DESCRIPCION:ChDir cambia el directorio actual a S.

## Chr

SINTAXIS: function Chr(l:integer);

DESCRIPCION:Chr regresa el caracter ASCII que corresponde a l.

## Circle [GRAPH UNIT]

SINTAXIS: procedure circle(x,y:integer; Radius:word);

DESCRIPCION:Circle dibuja un círculo con un radio de Radius alrededor de la coordenada x:y.

## ClearDevice [GRAPH UNIT]

SINTAXIS: procedure ClearDevice;

DESCRIPCION:ClearDevice borra la pantalla de las gráficas.

## ClearViewPort [GRAPH UNIT]

SINTAXIS: procedure Clear View Port;

DESCRIPCION:Borra la pantalla actual.

## Close

SINTAXIS: procedure Close (var F : File);

DESCRIPCION:Close drena el buffer del archivo F y entonces lo cierra.

## CloseGraph [GRAPH UNIT]

SINTAXIS: procedure CloseGraph;

DESCRIPCION:Reestablece el video en el modo anterior al modo gráfico actual. El procedimiento tambien libera la memoria usado por el sistema de gráficos.

## ClrEol [CRT UNIT]

SINTAXIS: procedure ClrEol;

DESCRIPCION:Limpia la pantalla desde la posición del cursor al final de la línea en la pantalla.

## ClrScr [CRT UNIT]

SINTAXIS: procedure ClrScr;

DESCRIPCION:Limpia la pantalla y posiciona el cursor en la esquina superior izquierda (localización 1,1).

## Concat

SINTAXIS: function Concat (S1,S2,...,Sn) : string;

DESCRIPCION:Combina cualquier número de strings y los regresa como uno solo. Si la longitud de la concatenación del string es mayor que 255, Turbo Pascal genera un Run-Time error.

## Copy

SINTAXIS: function Copy (S : string; P,L : integer) : string;

DESCRIPCION:Regresa una porción del string S el cual empieza con el caracter número P y tiene L caracteres.

## Cos

SINTAXIS: function Cos (R : real) : real;

DESCRIPCION:Regresa el coseno de R.

## Cseg

SINTAXIS: function Cseg : word;

DESCRIPCION:Regresa la dirección del segmento del código del programa.

## Dec

SINTAXIS: procedure Dec (var X : escalar; N : longint);

DESCRIPCION:Decrementa la variable X por N y si se omite N, X se decrementara en uno.

## Delay [CRT UNIT]

SINTAXIS: procedure Delay (MS : word);

DESCRIPCION:Detiene la ejecución del programa por MS milisegundos.

## Delete

SINTAXIS: procedure Delete (S : string; P,L : integer);

DESCRIPCION:Quita los L caracteres desde el string S empezando con el caracter número P.

## DelLine [CRT UNIT]

SINTAXIS: procedure DelLine;

DESCRIPCION:Borra la línea de la pantalla en donde el cursor esta localizado. Las líneas que se encuentren abajo de ésta, se recorren.

## DetectGraph [GRAPH UNIT]

SINTAXIS: procedure DetectGraph (var Gd,Gm : integer);

DESCRIPCION:Regresa el driver del tipo de gráficos detectado (Gd) y el modo gráfico (Gm) para la instalación adaptada.

## DiskFree [DOS UNIT]

SINTAXIS: function DiskFree (Drive : word) : longint;

DESCRIPCION:Regresa la cantidad de espacio libre en disco en bytes en el drive (0 = default, 1 = A, 2 = B, etc).

## DiskSize [DOS UNIT]

SINTAXIS: function DiskSize (Drive : word) : longint;

DESCRIPCION:Regresa el tamaño en bytes del disco en el drive (0 = default, 1 = A, 2 = B, etc).

## Dispose

SINTAXIS: procedure Dispose (P : pointer);

DESCRIPCION:Libera la memoria del HEAP en donde se localiza la variable pointer.

## DosExitCode [DOS UNIT]

SINTAXIS: function DosExitCode : word;

DESCRIPCION: Regresa el código de salida de un subproceso (0 = terminación normal, 1 = terminación por Ctrl-C, 2 = terminación por error de dispositivo, 3 = terminación por procedimiento Keep).

## DosVersion [DOS UNIT]

SINTAXIS: function DosVersion : word;

DESCRIPCION: Regresa la versión del sistema operativo. El dígito mayor en el byte más significativo y el menor en el menos significativo.

## DrawPoly [GRAPH UNIT]

SINTAXIS: procedure DrawPoly (NumPoints : word; var PolyPoints);

DESCRIPCION: Dibuja un polígono definido por el número de puntos NumPoints. El array PolyPoints contiene las coordenadas de los puntos del polígono.

## Dseg

SINTAXIS: function Dseg : word;

DESCRIPCION: Dseg regresa la dirección del segmento de datos del programa.

## EnvCount [DOS UNIT]

SINTAXIS: function EnvCount : integer;

DESCRIPCION: Regresa el número de strings definidas en el ambiente del DOS.

## EnvStr [DOS UNIT]

SINTAXIS: function EnvStr (i : integer) : string;

DESCRIPCION: Regresa el string número i del ambiente DOS.

## Eof

SINTAXIS: function Eof (F : File) : boolean;

DESCRIPCION: Regresa TRUE cuando el apuntador del archivo F encuentra el final de este.

## Eoln

SINTAXIS: function Eoln(F : File) : boolean;

DESCRIPCION: Regresa TRUE cuando el apuntador del archivo F encuentra el final de una línea (marcado con los caracteres de retorno de carro y alimentador de línea) o el final del archivo.

## Erase

SINTAXIS: procedure Erase (F : File);

DESCRIPCION: Borra de disco el archivo F y quita la información del directorio.

## Exec [DOS UNIT]

SINTAXIS: procedure Exec (Path, CmdLine : string);

DESCRIPCION: Ejecuta el archivo especificado en Path con los parámetros de línea definidos en CmdLine.

## Exit

SINTAXIS: procedure Exit;

DESCRIPCION: Causa la terminación abrupta del bloque que está siendo ejecutado.

## Exp

SINTAXIS: function Exp (R : real) : real;

DESCRIPCION: Regresa el logaritmo base E de R.

## FExpand [DOS UNIT]

SINTAXIS: function FExpand (P : PathStr);

DESCRIPCION: Acepta el nombre del archivo P, y regresa el nombre completo con el Path incluyendo el drive.

## FilePos

SINTAXIS: function FilePos (F : File) : integer;

DESCRIPCION: Regresa el número de registro hacia donde apunta el apuntador del archivo F.

## FileSize

SINTAXIS: function FileSize (F : File) : integer;

DESCRIPCION: Regresa el número de registros que contiene F.

## FillChar

SINTAXIS: procedure FillChar (Variable : tipo; I, Code : escalar);

DESCRIPCION: Llena los I bytes de memoria con el valor del código Code, el cual puede ser de cualquier tipo escalar, empezando con la dirección de la variable.

### FillEllipse [GRAPH UNIT]

SINTAXIS: procedure FillEllipse (x,y : integer; Xradio,Yradio : word);

DESCRIPCION:Dibuja un elipse centrado en las coordenadas x,y con un radio vertical Yradio y un radio horizontal Xradio. El elipse se rellena con el color y estilo de llenado actuales y el borde con el color actual.

### FillPoly [GRAPH UNIT]

SINTAXIS: procedure FillPoly (NumPoints : word; var PolyPoints);

DESCRIPCION:Dibuja un polígono con un número de puntos NumPoints. El array PolyPoints contiene las coordenadas para cada uno de los puntos del polígono.

### FindFirst [DOS UNIT]

SINTAXIS: procedure FindFirst(Path:string;Attr :word;var S: SearchRec);

DESCRIPCION:Regresa información en el primer archivo encontrado en el directorio PATH que tiene el atributo ATRR.Los valores estandar de ATRR son:

Const	
ReadOnly	= \$01;
Hidden	= \$02;
SysFile	= \$04;
Volumeld	= \$08;
Directory	= \$10;
AnyFile	= \$3f;

El tipo de dato SearchRec es definido como

```
Type
SearchRec = Record
    Fill : Array [1..21] of byte;
    Attr : byte;
    Time : longint;
    Size : longint;
    Name : string[12];
End;
```

Si la búsqueda es exitosa, el valor de DosError será cero.

### FindNext [DOS UNIT]

SINTAXIS: procedure FindNext(Var S : SearchRec);

DESCRIPCION:Regresa información en el siguiente archivo encontrado en el directorio,PATH definido en FindFirst .Si la búsqueda es exitosa,el valor de DosError será cero.

### Floodfill [GRAPH UNIT]

SINTAXIS: procedure FloodFill(x,y : integer; Border : word);

DESCRIPCION : Rellena y encierra el área de despliegue de gráficas con el color y modelo común. El área debe ser completamente encerrada por el color Border, y las coordenadas x:y deben posicionarse con el área a llenar.

### Flush

SINTAXIS: procedure Flush(Var F : Text);

DESCRIPCION:Manda al disco toda la salida del buffer al archivo F.

### Frac

SINTAXIS: function Frac(R:real):real;

DESCRIPCION:Regresa la porción no entera de R.

### FreeMem

SINTAXIS: procedure FreeMem(Var P :Pointer,l: integer);

DESCRIPCION:Libera l bytes de la memoria HEAP asociada con la variable P, que ya fué previamente localizada por GetMem.

### FSearch [DOS UNIT]

SINTAXIS: function FSearch(Path :PathStr;DirList : string): PathStr;

DESCRIPCION:Busca en la lista de directorios incluido en el DirList para un nombre de archivo que une el PATH. Si el archivo se encontró, el resultado regresará como un string.Si no se encontró, la función regresa un string vacío.

### FSplit [DOS UNIT]

SINTAXIS: procedure FSplit(Path : PathStr;Var Dir :Dirstr;Var Name : NameStr;Var Ext : ExtStr);

DESCRIPCION:Accepta un nombre de archivo PATH y regresa sus componentes. Se usan los siguientes tipos.

Type	
PathStr	= string[79];
DirStr	= string[67];
NameStr	= string[8];
ExtStr	= string[4];

### GetArcCoords [GRAPH UNIT];

SINTAXIS: procedure GetArcCoords(Var ArcCoords :ArcCoordsType);

DESCRIPCION:Regresa las coordenadas mas recientes que se usaron en los comandos Arc o Ellipse.La estructura es como se muestra a continuaci3n:

```
type
  ArcCoordsType = record
    y          : integer;
    Xstart,Ystart : integer;
    Xend,Yend  : integer;
  end;
```

### GetAspectRatio [GRAPH UNIT]

SINTAXIS: procedure GetAspectRatio(Var Xasp,Yasp :word);

DESCRIPCION:Regresa en Xasp y Yasp la resoluci3n efectiva de la pantalla de gráficas.El radio es calculado como Xasp dividido entre Yasp.

### GetBkColor [GRAPH UNIT]

SINTAXIS: function GetBkColor:word;

DESCRIPCION:Regresa el color de fondo de la paleta actual.

### GetCBreak [DOS UNIT]

SINTAXIS: procedure GetCBreak(Var Break:Boolean);

DESCRIPCION:Regresa el estado actual del CTRL-BREAK checado en el DOS. Cuando el Break es FALSE, el DOS checa el CTRL-BREAK mediante la consola,impresora, o I/O. Cuando es TRUE, el DOS checa el CTRL-BREAK en cualquier sistema llamado.

### GetColor [GRAPH UNIT]

SINTAXIS: function GetColor :word;

DESCRIPCION:Regresa el color del dibujo en el modo gráfico.

### GetDate [DOS UNIT]

SINTAXIS: procedure GetDate (Var Year,Month,Day,DayOfWeek :word);

DESCRIPCION:Regresa la fecha determinado por el reloj del sistema.

### GetDefaultPalete [GRAPH UNIT]

SINTAXIS: procedure GetDefaultPalette (Var Pal: PaletteType);

DESCRIPCION:Regresa en Pal la paleta por default para el drive de gráficas.La estructura del Palette Type es como sigue:

```
type
  PaletteType = record
    Size  : byte;
    Colors : array [0..MaxColor]of shortint;
  end;
```

### GetDir

SINTAXIS: procedure GetDir(d:byte;Var s: string);

DESCRIPCION:Trae el directorio del drive especificado por d.El directorio lo regresa en s. Si d es cero, GetDir busca el drive actual.

### GetDriverName [GRAPH UNIT]

SINTAXIS: function GetDriverName :string;

DESCRIPCION:Regresa el nombre del driver de gráficas.

### GetEnv [DOS UNIT]

SINTAXIS: function GetEnv(EnvVar :string): string;

DESCRIPCION:Regresa el string a la varaible especificada en EnvVar.

### GetFAttr [DOS UNIT]

SINTAXIS: procedure GetFAttr(Var F;Var Attr:word);

DESCRIPCION:Regresa el atributo para el archivo F.Antes de llamar a este procedimiento,F debe ser asignado pero no abierto.

### GetFillPattern [GRAPH UNIT]

SINTAXIS: procedure GetFillPattern(Var FP : FillPatternType);

DESCRIPCION:Regresa en FP la definici3n del modelo. La estructura del FillPatternType es :

```
Type
  FillPatterntype = Array [1..8] of byte;
```