

```

{***** M F 2 P *****)
{
*-----*
* Task : Demonstrates key read from MF-II keyboards. *
*-----*
* Author : Michael Tischer *
* Developed on : 01/28/92 *
* Last update : 01/28/92 *
{*****}

program MF2P;

uses Dos, Crt; { Add DOS and CRT units }

const CR = #13#10; { Carriage Return & Linefeed }

{*****}
{ * HexByte : Changes a byte into a two-digit hex string. * }
{ * Input : BVAL = Byte to be converted * }
{ * Output : Two-digit hex string * }
{*****}

function HexByte( bval : byte ) : string;

const HexDigits : array [0..15] of char = '0123456789ABCDEF';

var dummy : string[2]; { Get string }

begin
    dummy[0] := chr(2); { String consists of two characters }
    dummy[1] := HexDigits[ bval shr 4 ]; { Convert both }
    dummy[2] := HexDigits[ bval and $0F ]; { nibbles to hex }
    HexByte := dummy;
end;

{*****}
{ * TestMF: Tests whether the extended BIOS functions for reading the * }
{ * MF-II keyboard are available. * }
{ * Input : None * }
{ * Output : TRUE if the functions are available, otherwise FALSE * }
{*****}

function TestMF : boolean;

var Regs : Registers; { Processor registers for interrupt call }

begin
    Regs.AX := $1200; { Extended status function for MF-II keyboards }
    intr( $16, Regs );
    TestMF := ( Regs.AX <> $1200 ); { AX=$1200 : Function absent }
end;

{*****}
{ * GetMFKey : Reads a key using extended keyboard function 10H. * }
{ * Input : None * }
{ * Output : The returned keycode * }
{*****}

function GetMFKey : word;

var Regs : Registers; { Processor registers for interrupt call }

begin
    Regs.AH := $10; { Extended read function for MF-II keyboards }
    intr( $16, Regs );
    GetMFKey := Regs.AX; { Return keycode }
end;

{*****}
{ * MAIN PROGRAM * }
{*****}

var pdkey : word;

begin
    clrscr;
    writeln( 'MF2P - (c) 1992 by Michael Tischer' + CR );
    if ( TestMF ) then
        begin
            writeln( 'BIOS functions implemented for ' +
                'MF-II keyboards.' + CR + CR + 'Press any key ' +
                'or combination to display key codes.' + CR + CR +
                'Press <Esc> to end the program.' + CR );

            repeat { Input loop }
                pdkey := GetMFKey; { Get key }
            until pdkey = $1F;
        end;
    else
        writeln( 'MF-II keyboard not available.' + CR );
    end;
end;

```

```
write( 'Scan : ', HexByte(hi(pdkey)), ' ',  
      'ASCII: ', HexByte(lo(pdkey)) );  
if ( (lo(pdkey) = $E0) and (hi(pdkey) <> 0 ) ) then  
  write( ' <---- MF-II key' );  
  writeln;  
until ( pdkey = $011b );      { Repeat until user presses <ESC> }  
writeln( CR );  
end  
else  
  writeln( 'No BIOS extensions available for MF-II keyboards!' );  
end.
```