

```

{*****}
{*              D V I P              *}
{*-----*}
{* Task          : Demonstrates direct access to video RAM. *}
{*-----*}
{* Author        : Michael Tischer *}
{* Developed on   : 01/02/87 *}
{* Last update    : 02/26/92 *}
{*****}

```

```
program DVIP;
```

```
Uses Crt, Dos;                                { Add CRT and DOS units }
```

```

const NORMAL      = $07;      { Define character attributes }
  HIINT           = $0f;      { on monochrome video card }
  INVERSE         = $70;
  UNDERSCORED    = $01;
  BLINKING        = $80;

```

```

  BLACK          = $00;      { Color attributes on color video card }
  BLUE           = $01;
  GREEN          = $02;
  CYAN           = $03;
  RED            = $04;
  VIOLET         = $05;
  BROWN         = $06;
  LGHTGRAY       = $07;
  DARKGRAY       = $01;
  LGHTBLUE       = $09;
  LGHTGREEN      = $0A;
  LGHTCYAN       = $0B;

```

```

    LGHTRED      = $0C;
    LGHTVIOLET   = $0D;
    YELLOW       = $0E;
    WHITE        = $0F;

```

```
type TextType = string[80];
```

```
var VSeg : word; { Segment address of video RAM }
```

```

{*****}
{ * InitDPrint: Gets the segment address for DPrint. * }
{ * Input      : None * }
{ * Output     : None * }
{*****}

```

```
procedure InitDPrint;
```

```
var CRTC_PORT : word absolute $0040:0063; { Seg.addr.: BIOS var.reg. }
```

```
begin
```

```

    if CRTC_PORT = $3B4 then { Monochrome adapter? }
        VSeg := $B000 { Yes --> Video RAM at B000:0000 }
    else { No --> Must be a color adapter }
        VSeg := $B800; { Video RAM at B800:0000 }

```

```
end;
```

```

{*****}
{ * DPrint: Writes a string directly to video RAM. * }
{ * Input   : - COLUMN: The display column * }
{ *          - SCROW : The display row * }
{ *          - DCOLR : Character color (attribute) * }
{ *          - STROUT: The string to be displayed * }

```

```
{* Output : None *}
{*****}
```

```
procedure DPrint( Column, ScRow, DColr : byte; StrOut : TextType);
```

```
var PAGE_OFS : word absolute $0040:$004E;{ Seg. addr: BIOS var. reg. }
    Offset    : word;           { Pointer to current display position }
    i, j      : byte;           { Loop counter }
    Attribute : word;           { Display attribute }
```

```
begin
```

```
    Offset := ScRow * 160 + Column * 2 + PAGE_OFS;
    Attribute := DColr shl 8; { High byte for word access to video RAM }
    i := length( StrOut );    { Get string length }
    for j:=1 to i do          { Execute string }
        begin                 { Apply next character attribute to video RAM }
            memw[VSeg:Offset] := Attribute or ord( StrOut[j] );
            Offset := Offset + 2; { Set to next ASCII attribute pair }
        end;
    end;
```

```
end;
```

```
{*****}
{* Demo: Demonstrates DPrint routine. *}
{* Input : None *}
{* Output : None *}
{*****}
```

```
procedure demo;
```

```
var Column,                               { Current display position }
    ScRow,
    DColr : integer;
```

