

Procedure Hardcopy;

| (c) 1987 AHSoft |

var xp,yp,x0,y0,x,y: integer;
vrstic: byte;

Function Test_tocke(a,b: integer): byte; { Test_tocke }
var bit: byte;

begin
inline (\$F5/\$C5/\$D5/\$E5/
\$ED/\$5B/a/\$2A/b/
\$CD/\$F0/\$BB/\$32/bit/
\$E1/\$D1/\$C1/\$F1);
if bit > 1 then bit := 1;
Test_tocke := bit
end; { Test_tocke }

Procedure Copy_byte(xzac,xkon: integer; stevilo: byte; var y: integer;
var x: integer;
tocke,a,b: byte;

begin
for x := xzac to xkon do begin
b := 0;
for tocke := 1 to stevilo do begin
a := Test_tocke(x,y);
b := b shl 1;
b := a or b;
y := y-2
end; { for tocke }
write(lst,chr(b));
y := y+2*stevilo
end; { for x }
end; { Copy_byte }

begin { Hardcopy }
inline (\$F5/\$D5/\$E5/
\$CD/\$C6/\$BB/
\$ED/\$53/xp/\$22/yp/
\$E1/\$D1/\$F1); { GET CURSOR }
inline (\$D5/\$E5/\$CD/\$CC/\$BB/
\$ED/\$53/x0/\$22/y0/
\$E1/\$D1); { GET ORIGIN }
inline (\$F5/\$C5/\$D5/\$E5/
\$11/\$00/\$00/\$21/\$00/\$00/
\$CD/\$C9/\$BB/
\$E1/\$D1/\$C1/\$F1); { SET ORIGIN 0,0 }
writeln(lst);
write(lst,chr(27),chr(51),chr(21)); { LF 7/72" }
y := 398;
vrstic := 28;
while vrstic > 0 do begin
writeln (lst,chr(13)); { CR }
write(lst,chr(27),chr(76),chr(64),chr(1)); { double bit image 320 }
Copy_byte(0,319,7,y);
write(lst,chr(27),chr(76),chr(64),chr(1));
Copy_byte(320,639,7,y);
y := y-14;
vrstic := vrstic-1
end; { while }
write(lst,chr(27),chr(51),chr(12)); { LF 4/72" }
writeln(lst,chr(13));
write(lst,chr(27),chr(76),chr(64),chr(1)); { double bit image 320 }
Copy_byte(0,319,4,y);
write(lst,chr(27),chr(76),chr(64),chr(1));
Copy_byte(320,639,4,y);
inline (\$F5/\$C5/\$D5/\$E5/
\$ED/\$5B/x0/\$2A/y0/
\$CD/\$C9/\$BB/
\$E1/\$D1/\$C1/\$F1); { SET ORIGIN prvotni }
inline (\$F5/\$C5/\$D5/\$E5/
\$ED/\$5B/xp/\$2A/yp/
\$CD/\$C0/\$BB/
\$E1/\$D1/\$C1/\$F1); { SET CURSOR prvotni }
write(lst,chr(27),chr(51),chr(36)); { LF 1/6" }
writeln(lst,chr(13))
end; { Hardcopy }