

```

/*****
/*
/*----- C O N F I G C -----*/
/* Task : Displays the configuration of the PC. */
/*-----*/
/* Author : Michael Tischer */
/* Developed on : 08/13/87 */
/* Last update : 04/07/95 */
/*-----*/
/* Memory model : SMALL */
/*****
/*== Add include files =====*/

#include <dos.h>
#include <stdio.h>

/*== Type definitions =====*/

typedef unsigned char BYTE; /* Create a byte */

/*== Macros =====*/

#ifdef MK_FP
    #undef MK_FP
#endif

#ifdef peekb
    #undef peekb
#endif

#define MK_FP(seg, ofs) ((void far *) ((unsigned long) (seg)<<16|(ofs)))
#define peekb(seg, ofs) *((BYTE far *) MK_FP(seg, ofs))

/*== Constants =====*/

#define TRUE ( 0 == 0 ) /* Constants make reading */
#define FALSE ( 0 == 1 ) /* program code easier */

/*****
/* CLS: Clears current screen and places cursor in upper-left corner.*/
/* Input : None */
/* Output : None */
/*****

void Cls( void )
{
    union REGS Register; /* Register variables for interrupt call */

    Register.h.ah = 6; /* Function number for scroll up */
    Register.h.al = 0; /* 0 = Clear */
    Register.h.bh = 7; /* White text on black background */
    Register.x.cx = 0; /* Upper-left corner of screen */
    Register.h.dh = 24; /* Bottom-right screen */
    Register.h.dl = 79; /* coordinates */
    int86(0x10, &Register, &Register); /* Call BIOS video interrupt */

    Register.h.ah = 2; /* Function number for Set cursor position */
    Register.h.bh = 0; /* Screen page 0 */
    Register.x.dx = 0; /* Upper-left screen coordinates */
    int86(0x10, &Register, &Register); /* Call BIOS video interrupt */
}

/*****
/* PRINTCONFIG: Displays PC configuration. */
/* Input : None */
/* Output : None */
/* Info : Configuration varies with the type of PC */
/*****

void PrintConfig( void )
{
    union REGS Register; /* Register variables for interrupt call */
    BYTE AT; /* AT or higher? */

    Cls(); /* Clear screen */

```

```

AT = (peekb(0xF000, 0xFFFF) == 0xFC);
printf("CONFIG - (c) 1987, 92 by Michael Tischer\n\n");
printf("Your PC Configuration \n");
printf("-----\n");
printf("PC type          : ");

switch( peekb(0xF000, 0xFFFF) )      /* Read PC type and display */
{
    case 0xFF : printf("PC\n");          /* 0xFF (FFH) is a PC */
                break;
    case 0xFE : printf("XT\n");          /* 0xFE (FEH) is an XT */
                break;
    default   : printf("AT or higher\n"); /* 0xFC (FCH) is an AT */
                break;
}
printf("Conventional RAM      : ");
int86(0x12, &Register, &Register);    /* RAM from BIOS interrupt */
printf("%d K\n", Register.x.ax);        /* Display RAM */
if ( AT )                               /* Is the PC an AT? */
{                                       /* Yes */
    Register.h.ah = 0x88; /* Read function number for extended memory */
    int86(0x15, &Register, &Register); /* Get RAM size */
    printf("Additional RAM      : %d K over 1 megabyte\n", Register.x.ax);
}
int86(0x11, &Register, &Register);    /* Call BIOS configuration */
printf("Default video mode    : ");    /* interrupt */
switch(Register.x.ax & 48)
{
    case 0 : printf("Undefined\n");
                break;
    case 16 : printf("40x25 character color card\n");
                break;
    case 32 : printf("80x25 character color card\n");
                break;
    case 48 : printf("80x25 character mono card\n");
                break;
}
printf("Disk drives          : %d\n", (Register.x.ax >> 6 & 3) + 1);
printf("Serial interfaces     : %d\n", Register.x.ax >> 9 & 0x03);
printf("Parallel interfaces   : %d\n\n", Register.x.ax >> 14);
}

/*****
**                               **
MAIN PROGRAM
**
*****/

void main()
{
    PrintConfig();                /* Display configuration */
}

```