How To Tell if MacsBug is Installed

```
/*
        How to tell if MacsBug is Installed
        This is a small snippet of code that can be used to to detect if
        macsbug is installed or not. NOTE:  This code is intended to only work with
        version 6.2 of macsbug.  You should refer to your Low Level Debugger's manual
        for more information on how they install themselves.

        This code is based on information obtained from the MacsBug Reference.
        The basic assumptions are that macsbug will install itself in the following
        manner:

        If you are running in 24 bit mode, then the high -order byte of MacJmp is a flags
        byte that contains the following information:

        Bit     Meaning
        ---     -------------------------------------
        7       -  Set if debugger is running
        6       -  Set if debugger can handle system errors
        5       -  Set if debugger is installed
        4       -  Set if debugger can support discipline utility

        The lower 3 bytes are used to store the address of the debugger's entry point.

        If you are running in 32-bit mode, the flags byte is moved to address 0xBFF and
        the long word at MacJmp becomes a full 32-bit address that points to the
        debugger's entry point..

        ADDENDUM:  The above information seems to be incorrect in the reference
        manual. I have found through testing etc. that in both modes, the Flag Byte
        appears at location 0xBFF.  The code reflects these findings.

*/


#include <stdio.h>
// Assumes inclusion of <MacHeaders>


// Prototypes
Boolean MacsBugInstalled(void);

#define BYTEMASK   0x20   // Used to detect if bit 5 is set


Boolean MacsBugInstalled()
{
    Ptr FlagByte  = (Ptr)0xBFF;   // This is used only if running in 32 bit mode

    return (*FlagByte & BYTEMASK);

}

main()
{
    if (MacsBugInstalled())
            printf ("MacsBug is installed, go ahead and crash!");
```

```
    else
            printf ("MacsBug is not installed, be careful!");
}
```