```
#include "supp.h"

void displaySpeed(int speed)
{
//convert speed to char output
    char text[3];
    intToChar(speed, text);
    //WinDrawChars(text,3,0,40);


    FieldType *fldP;
    FormType* frmP;

    frmP = FrmGetActiveForm();

    //char *strBuffer;

    //strBuffer = (char *) MemPtrNew(3);
    //StrCopy(strBuffer, text);


    fldP = (FieldType *) FrmGetObjectPtr(frmP,
                          FrmGetObjectIndex(frmP, MainSpeedField));
    FldFreeMemory(fldP);   // initialize everything, just in case.
    FldSetMaxChars(fldP, StrLen(text));
    FldSetTextPtr(fldP, text);
    FldRecalculateField(fldP, true);
    FldDrawField(fldP);
}

void displayDTCs(int numDTCs)
{
    char text[3];
    intToChar(numDTCs, text); //convert numDTCs to text

    FieldType *fldP;
    FormType* frmP;

    frmP = FrmGetActiveForm();
    fldP = (FieldType *) FrmGetObjectPtr(frmP,
                          FrmGetObjectIndex(frmP, MainDTCsField));
    FldFreeMemory(fldP);   // initialize everything, just in case.
    FldSetMaxChars(fldP, StrLen(text));
    FldSetTextPtr(fldP, text);
    FldRecalculateField(fldP, true);
    FldDrawField(fldP);
}

void intToChar(int speed, char* str)
{
    //assuming its no higher than 999
```

```
    //add 48 to get ascii val
    str[0]=speed/100      +48;
    if (str[0]==48) str[0]=32;  //if speed<100, display space instead of 0
    str[1]=(speed%100)/10 +48;
    str[2]=(speed%100)%10 +48;
    str[3]=NULL;
}
```