**修正後プログラム(int型からchar\*型へ変換する関数のみ変更)**

char \*num2str(int i)

{

static int num2str\_num=0;

static char str[100][30];

if(num2str\_num>=100)num2str\_num=0;

sprintf(str[num2str\_num],"%d",i);

num2str\_num++;

return str[num2str\_num-1];

}

**元のプログラムの問題点とその解決法**

アドレスが返り値になっている文字配列 char str[30] は関数内で定義されているため、関数が終了後破棄される。するとアドレスを返したところでそこにはないもないことになり、エラーが出される。

ここで定義時に static char とすることで関数が終了後も変数が所持されるようにする。この段階で実行すると文字列をprintfするときに同じ文字が二回帰ってくる。これは、同じアドレスにprintfで返すときに書き込んでいるせいでおきるので、毎回保存する領域を変更するために文字配列自体を配列にして、書き込むアドレスを毎回変化させるようにした。このプログラムにおいてアドレスを変化させることは有限回しかできないので、上限回に達したら最初に戻すというプログラムにした。