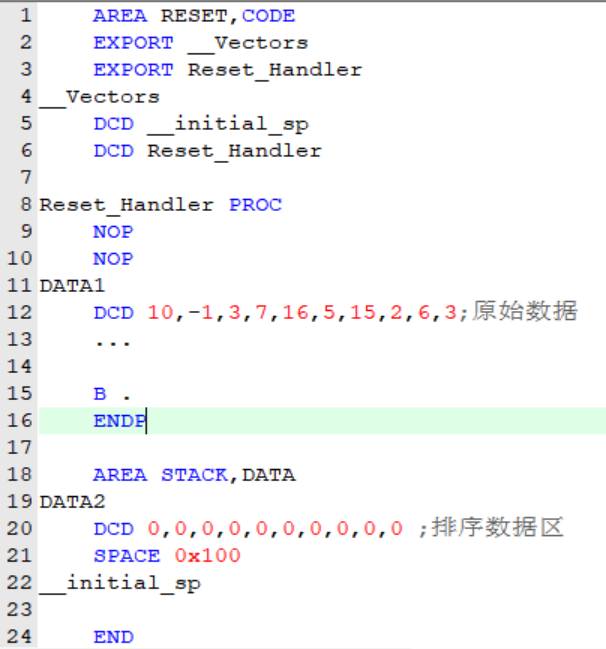
**微处理器**

一、要求：



如上图，将DATA1数组中存放的10个数，从小到大排序并存放到DATA2数组中。要求画出流程图，编写代码并给出注释；提交利用Keil软件执行后，DATA2数组在存储器中数据的结果截图。

二、做题思路（数据设计—流程设计—汇编流程设计）：

考虑到DATA1使定义在CODE区（0x00000000~0x1FFFFFFF）的变量，其地址为只读模式，我们无法写入并更改。因此，我选择利用一个循环先将DATA1的十个数字输出到可编写的DATA区(0x20000000~0x3FFFFFFF)，在这里考虑到DATA2的地址0x20000000那么十个DCD数据的位置0x20000028。因此，只要在0x20000028之后选取位置编写DATA1，就不会产生干扰。因此，我们在此处选择了0x20000100来记录DATA1的数据。

模块一：DATA1向地址0x20000100输出数据，代码如下：

图示, 示意图

中度可信度描述已自动生成

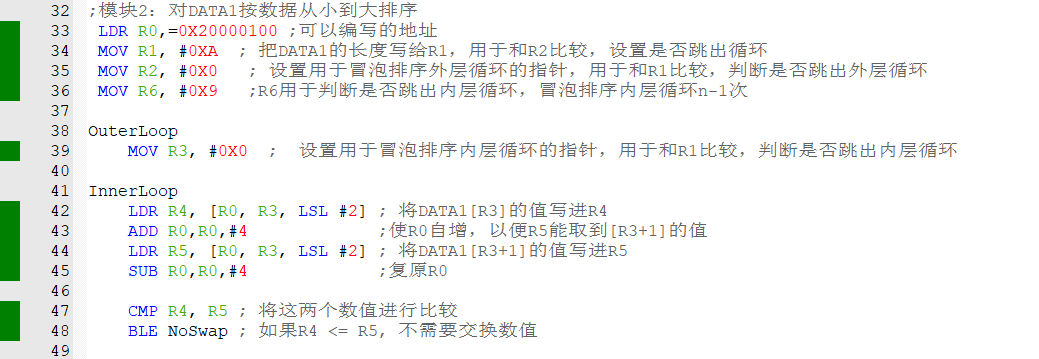
模块一实现的效果如图所示：

图片包含 表格

描述已自动生成

接着，对0x20000100地址上的十个数据按从小到大进行排序，大一学习C语言时，我们最熟悉的排序方式即为冒泡排序，因此在这里我选择利用冒泡排序来完成。需要注意的是，冒泡排序内层只循环n-1次，外层循环n次。

模块二：冒泡排序实现从小到大排列，代码如下



图片包含 日程表

描述已自动生成

模块二实现的效果如下：

表格

描述已自动生成

最后，我们只需要利用循环将地址为0x20000100的是个数据写入DATA2即可。DATA2位于可读写的DATA区，故无需考虑读写问题。

模块三：使用循环将排序后的数据写入DATA2，代码如下：

图片包含 日程表

描述已自动生成

模块三实现的效果如下：

表格

中度可信度描述已自动生成

三、流程图设计（见下页）

图片包含 表格

描述已自动生成

附件：汇编源码&运行结果

汇编源码：

文本

描述已自动生成

图片包含 日程表

描述已自动生成

图片包含 文本

描述已自动生成

文本

描述已自动生成

运行结果：

表格

中度可信度描述已自动生成