

微嵌第 7 章参考答案

7.3 编写一个完整 ARM 汇编程序实现如下功能：当 $R3 > R2$ 时，将 $R2 + 10$ 存入 $R3$ ，否则将 $R2 + 100$ 存入 $R3$ 。

```
AREA ADDITION, CODE, READONLY
ENTRY
MOV R2, #1 ;设置 R2 的值
MOV R3, #2 ;设置 R3 的值
CMP R3, R2
BHI GREATER
ADD R3, R2, #100
B STOP
GREATER
ADD R3, R2, #10
STOP
END
```

7.4 将数据段中 10 个数据中的偶数个数统计后放入 $R0$ 寄存器。

```
AREA BUF, DATA, READONLY
Array DCD 12, 23, 34, 45, 56, 67, 78, 89, 19, 58
AREA EVENNUMBER, CODE, READONLY
ENTRY
LDR R1, =Array
MOV R3, #10 ; 将 R3 用作计数器
MOV R0, #0 ; R0 用于统计偶数个数
START LDR R2, [R1]
TST R2, #1 ;按位与 0b0001, 奇偶校验
BEQ EVEN ;偶数最低位为 0, 与上述掩膜按位与后 Z=1
ODD ADD R1, R1, #4
```

```

        SUB R3,R3,#1
        CMP R3,#0
        BNE START
        BEQ STOP
EVEN ADD R0,R0,#1
        ADD R1,R1,#4
        SUB R3,R3,#1
        CMP R3,#0
        BNE START
        BEQ STOP
STOP
        END

```

7.6 试编写一个循环程序，实现 1 至 100 的累加。

```

        AREA SUM,CODE,READONLY
        ENTRY
        MOV R0,#100
        MOV R1,#0
LOOP ADD R1,R1,R0           ;结果存放在 R1
        SUBS R0,R0,#1
        CMP R0,#0
        BNE LOOP
        END

```

7.9 编写完整汇编程序调用 C 函数计算 N! (N<=10)。

```

        PRESERVE8           ;声明调用 C 代码时栈是 8 字节对齐的
        AREA FACTORIAL,CODE,READONLY
        ENTRY
        IMPORT factorial     ;声明外部引用符号 factorial
        MOV R0,#10
        BL factorial         ;调用 C 程序 factorial(),其结果从 R0 返回
        END

factorial.c
#include <stdio.h>

```

```

int factorial(int N) {
    int i = 1;
    int fac = 1;
    if (N == 0)
        return fac;
    for(i=1; i<=N; i++)
        fac = fac * i;
    return fac;
}

```

7.10 C 程序调用汇编函数计算字符串长度，并返回长度值。

注意这里是 C 调用汇编函数，不是内嵌汇编或者内联汇编

main.c

```

#include <stdio.h>
extern int StringLength(char *s);
int main() {
    char * str = "Hello world." ;
    int length = 0;
    length = StringLength(str);
    return 0;
}

```

汇编：

```

AREA STRINGLENGTH, CODE, READONLY
EXPORT StringLength

```

StringLength

```

MOV R1, R0          ;转移输入地址
MOV R0, #0          ;初始化 R0 以计数

```

LOOP

```

LDRB R2, [R1], #1    ;读取字符串的下一字符到 R2 寄存器
CMP R2, #0           ;判断是否到结尾，C 语言字符串结尾为 '/0'
ADDNE R0, R0, #1      ;未到结尾则+1 循环
BNE LOOP

```

MOV PC, LR

END