一、简答

1. **ARM9 指令系统**寻址方式有几种?
2. ARM9 支持几种处理器模式，都是什么？
3. ARM9 处理器有几种工作模式？都是什么？由 CPSR 中的什么标志位标识？
4. CPSR 中的 T 标识有什么作用，什么情况下会修改 T 值？
5. 给出两种以上修改 CPSR 中的 T 值的程序段？
6. 指令中快速中断模式下的 R13 如何书写？除了作为通用寄存器外还有什么作用？
7. R14，R15 寄存器有什么作用？
8. CPSR 中的 N、Z、C、V 都有什么标志？有什么意义？如何改变它们的值？
9. CPSR 中 I 是什么标志，有什么意义？给出改变 I 标志的指令？
10. CPSR 中 F 是什么标志，有什么意义？给出改变 F 标志的指令？
11. CPSR 中[M4:M0]是什么标志，有什么意义？在什么情况下此标志会改变？给出使用指令修改此标志的程序段？
12. 什么是异常？
13. ARM7 有几种异常，都是什么？
14. 发生复位异常时，程序将转移到什么地址去执行？
15. 在用户模式下，当指令预取中止异常与 IRQ 异常同时发生时，系统将首先响应那类异常？为什么？
16. 说明为什么从 IRQ 异常返回时使用指令 **SUB PC , R14_irq , #4？**说明为什么从数据中止异常返回时使用指令 **SUB PC , R14_abt , #8？**
17. 嵌入式系统有几种复位方式？
18. 存储 THUMB 指令需要几个字节单元，存储 ARM 指令需要几个字节单元？
19. 下边哪些是合法的立即数，为什么？

    0x00FF0000

    0x03FC0000

    0x07F80000

    0x09FC0000
20. 汇编指令中 cond 的作用是什么？汇编指令中 S 的作用是什么？

21. 说明下边指令的功能

MOVCC R0，#5

SUBGES R0,R0,R1

22. 满足下边条件时，CPSR 中对应标志位的值是什么？有什么含义？

EQ Z=?

VC C=?

23. 说明下列语句的寻址方式？

1) ADD R0，R1，＃5
2) MOV R0，R1
3) MOV R0，R1，LSL R3
4) STR R0，[R1]
5) LDR R0，[R1，＃5]

6) LDR R0，[R1,R2]

7) BEQ process1
.....
process1
.....

8) LDMIA R0，{R1-R5}

9) LDMIA R0，{R2-R6}；
STMIA R1，{R2-R6}；

10) STMFA R13!，{R0,R1}

24. 图示并说明堆栈指令 STMFA R13!，{R0-R6}的执行情况？

25. 图示并说明指令 LDMIA R13!，{R0-R6}的执行情况？

26. 给出将堆栈指针设置为 0x40003000 的汇编指令。

27. 给出使用多字拷贝指令将 0x40003000 开始的连续 176 个字节单元数据拷贝到 0x40008000 地址处的程序段？

28. 写出下面指令的执行结果。

1) MVN R0,＃20

2) LRD R0,=0x40001234
RSB R0,R0,#3

3) LDR R0,=0xA3BC3EC0;LDR R1,=0x0000FFFF; EOR R0,R0,R1

4) LDR R0,=0xA3BC5395;LDR R1,=0x0000CCCC; BIC R0,R0,R1

5) LDR R0,=0xA3BC5395;LDR R1,= 0xA3BC5365; 指令 TEQ R0,R1 执行后 CPSR 中 Z 的值是什么？

6) N=0，C=0，Z=1，V=0，F=0，I=0，T=1，[M4:M0]=010011, 指令 MRS R0,CPSR 执行后结果?

7) 给出使用 Pblock 代表寄存器列表 {r0-r3,r7,r5,r9}的指令？

29. MOV R0，#12
SUBS R0，R0，#26
程序段执行的结果是什么？CPSR 状态标志位 C=?

30. 说明指令 SMULL 的功能，并给出下边指令执行的结果？

```
LDR  R2，=0xFFFFFFF0
LDR  R3,=0x0000000C
SMULL R0,R1,R2,R3
```

31. 说明指令 SMLAL 的功能，并给出下边指令执行的结果？

```
LDR  R2，=0xFFFFFFF0
LDR  R3,=0x0000000C
 MOV R0,#5
 LDR R1,=0x40003000
 SMLAL  R0,R1,R2,R3
```

32. 说明指令 UMLAL 的功能，并给出下边指令执行的结果？

```
LDR  R2，=0xFFFFFFF0
LDR  R3,=0x0000000C
MOV R0,#5
LDR R1,=0x40003000
```

UMLAL R0,R1,R2,R3

33. 写出几种将寄存器 R0 清零的汇编指令?
34. 使用 AND 指令编写程序段，实现保留寄存器 R0 高 8 位，清除低 24 位?
35. 使用 ORR 指令编写程序段，实现将寄存器 R0 位 30、位 24、位 23、位 20、位 11、位 3、位 2 置 1，其它位不变?
36. 编写程序段，实现判断 R0 是否为 0?
37. 给出使用 BIC，将寄存器 R0 位 30、位 24、位 23、位 20、位 11、位 3、位 2 清零的指令？
38. 给出使用 TST 指令测试寄存器 R0 中位 20 是否为 0 的程序段？
39. 分别给出使用 EOR 指令 TEQ 指令都判断两个 32 位数是否相同的程序段，并说明两天指令的区别？
40. 转移指令 B Label； 说明 Label 取值范围？给出 PC 的计算公式？
41. 解释 BX R0 指令的功能，并说明 PC 和 CPSR 中 T 位的计算方法？
42. 已知[0x40003000]=0x11223344,R1=0x12345678,写出指令 SWP R0,R1,[R2]的执行结果?
43. CPSR 中，C=1，z=1,v=0,N=0,I=0,F=0,T=0,[M4:M0]=10011，指令 MRS R0,CPSR 执行后,R0 的值是什么？
44. 给出下边程序段编译后的结果，并说明$的作用？

```
LCLS    str1
LCLS    str2
str1    SETS    “book”
str2    SETS    “It is a $str1”
```

45. 给出调用编号为 5 的系统例程的程序段？
46. 举例说明汇编语言如何调用 C 变量、C 函数?说明 C 语言如何调用汇编函数，以及参数传递规则?
47. 定义一个宏，实现 SUB R0,R1,R2 的功能。并说明函数与宏的区别？写出宏调用指令？
48. 解释说明交叉调试方式？
49. 举例说明伪指令 EXPORT 的作用?
50. 举例说明 LTORG 的功能和使用条件？
51. 编译器对以下程序段进行编译，其中，伪指令 ADR R2，LOOP 的编译结果是？

```
LOOP    MOV     R1，#0xF0
        ADR     R2，LOOP
```

52. 编译器对以下程序段进行编译，其中，伪指令 ADRL R2，LOOP 的编译结果是？

```
LOOP    MOV     R1，#0xF0
        ADRL    R2，LOOP
```

53. 编译器对 LDR R0,=0x40003000 进行编译，编译的结果是什么？
54. 使用 PROC、ENDP 编写汇编函数，实现 C=A*B+D 的功能?
55. C 语言调用汇编函数时，参数是如何传递的？
56. R0 的内容为存储器地址，从 R0 指定的地址中读取一个字，并将 R0 的内容加 4，给出实现此功能的指令

57. 在汇编程序中可调用 C 库中函数_ printf，给出引用声明。

二、简单应用题

1. 说明下边函数的功能？

```
welcomefun      STMFD sp!,{lr}
                ADR r0,adrstrarm
                LDR r0,[r0,#0]
                BL   _printf
                LDMFD sp,{pc}
```

2. 说明以下函数的功能？

```
void my_strcpy(const char *src,char *dst)
{ int ch;
  __asm {
    loop:     LDRB ch,[src],#1;
              STRB ch,[dst],#1;
              CMP ch,#0;
              BNE loop;
}
}
```

3. 定义一个结构化内存表，表首地址 0x40003300, count 分量为 4 字节，x 分量为 8 字节，Y 分量为 4 字节
4. 用汇编语言实现 128 位数的减法。
5. 分析下面程序的功能。

```
STMFD SP!,{R0-R6}
LDR R6,=SRC
LDMIA R6!,{R0-R5}
LDR R6,=DST
STMIA R6!,{R0-R5}
LDMFD SP!,{R0-R6}
```

6. 已知 R1=0x30, R5=1,R6=2,R7=0x3FC ，执行

```
STMIA   R1!,{R7,R6,R5}
LDMDA R1!,{R5-R7}
```

R1,R5,R6,R7 的值分别是多少。

7. 分析下面程序的功能。

```
ADR R0, TTCODE+1
BX R0
```

8. 对于以下程序，写出执行 hello mymacro tom, R1,R0 的展开结果

```
MACRO
$lab    mymacro $var1,$var2, $var3
$ lab.loop1
BGE     $lab.loop1
```

```
$ lab .loop2
BL          $var1
BGT       $ lab.loop2
SUB   $var2, $var3，1
        MEND
```

9. 阅读下面的程序段，并回答 R0，R1，R2,以及 R3 中最终数值是多少？

```
Num EQU          0X10001FFF
    MOV     R0，#100
    LDR     R1，=Num
    STR     R0, [R1],-8
    ADD     R3,R0,R0,LSR #2
    RSB     R2,R0,R0,LSL #2
    STMIA   R1!,{R2,R3}
    LDMDA R1,{R0,R2,R3}
```

10. 说明下面程序段的功能,并给出 ClassNO,X 的地址：

```
    MAP       0x40001000
    NO        FIELD   4
    NAME      FIELD   4
    ClassNo   FIELD   4
    X         FIELD   8
```

11. 下面代码段的功能是根据输入参数（在 R0 中）进行程序散转，若 R0=0 则执行 DoAdd，若 R0=1 则执行 Dosub。请补充其中缺少的代码：

```
arithfunc                          ;
        CMP       r0, #num          ; 比较参数
        MOVHS     pc, lr            ; 若超出范围则程序返回
        _____________
        _____________
JumpTable
        DCD       DoAdd
        DCD       DoSub
DoAdd
        ADD       r0, r1, r2        ; =0 时的操作
        MOV       pc, lr            ; 返回
DoSub
        SUB       r0, r1, r2        ; =1 时的操作
        MOV       pc,lr             ; 返回
```

12. 用一条 ARM 指令分别以下完成：

（1）R0=R1/16

（2）R1=R2*3

（3）将字数据 0xFFFFFFFF 送入寄存器 R0

（4）将 R0 的第 0 位和第 2 位取反，其余位不变

13. 下面嵌入式汇编两个语句都有错误，请修改。

```
__asm
{
    MOV  R0,x
    ADD   y,R0,x/y
 }
```

14. R0 的内容为 0X0000，R1 的内容为 0X0001，[0x0001]=0x23,执行指令 LDR R0，[R1],#4 后 R0 和 R1 的值分别是多少？
15. R1 的内容为 0X2021，R2 的内容为 0XFE52，R3 的内容为 0X7C13，内存中从 0X7C00 开始以字为单位依次存放 1～100，请问执行语句 SWP R1,R2,[R3]后寄存器及存储单元的变化？
16. 修改程序错误

```
AREA addreg,CODE,READONLY
ENTRY
Main     ADR r0,ThumbProg
          BX   r0
 CODE16
 ThumbProg              ADR r0,ARMProg
                        BX    r0
 CODE32
 ARMProg       MOV R4,#4
               MOV R5,#5
               SUB  R4,R4,R5
 STOP          MOV R0,#0X18
               MOV R0,#20026
               SWI   #0xAB
 END
```

17. 说出以下两条指令的区别

```
PINSEL0 = 0x05 << 16;
PINSEL0 = (PINSEL0 & 0xFFF0FFFF) | (0x05 << 16);
```

18. 参考 CPSR 寄存器中各标志位的含义，使处理器工作在系统模式下。请补全程序中的操作码。

```
___        R0, CPSR
AND        R0, R0,#0xFFFFFFE0
ORR        R0, R0,#0x1F
___        CPSR_fsxc, R0
```

19. 下面代码实现什么功能

```
EOR   R1, R0, R0, ROR #16 ;
BIC   R1, R1, #0xFF0000
MOV   R0, R0, ROR #8
EOR   R0, R0, R1, LSR #8
```

20. 假设 R0 的内容为 0x8000，寄存器 R1、R2 的内容分别为 0x01 与 0x10，存储器内容为 0。连续执行下述指令后，说明每条指令执行后 PC 如何变化？存储器及寄存

器的内容如何变化？

STMIB R0!,{R1,R2}

LDMIA R0!,{R1,R2}

21. 下面的程序中，语句 IMPORT　strhello 的作用是什么？执行 main 过程后的结果是什么？

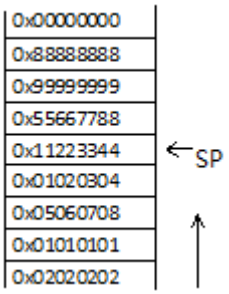
C 语言代码文件 str.c：

```
char *strhello="Hello world!\n\0";
```

汇编代码文件 hello.s

```
        AREA ||.text||, CODE, READONLY
main PROC
            STMFD       sp!,{lr}
            LDR         r0,=strtemp
            LDR       r0,[r0]
            BL          _printf
            LDMFD       sp!,{pc}
strtemp
  DCD         strhello
        ENDP
        EXPORT    main
        IMPORT    strhello
        IMPORT    _main
        IMPORT    _main
        IMPORT    _printf
        IMPORT    ||Lib$$Request$$armlib||, WEAK
        END
```

22. SP 及内存关系如图所示，SP=0x100，执行完 LDMIA SP！，{R0-R2}后 R0=?R1=?R2=?SP=?



23. 下面的指令实现什么功能？

MRS　R0，CPSR

ORR　R0，R0,0X1F

MSR　CPSR_c,R0

24. 下面程序实现什么功能？

CMP　R0，　#1

BLLT　　DOSUB1

BLGE　DOSUB2

25. 编写子程序实现将存储器中起始地址 S1 开始的 6 个字数据移动到 S2 处（要求使用 LDM 和 STM 语句）

26. 编写汇编程序段，实现数据 0x8A25B379 乘以 5 的功能。

27. 说明下面异常响应伪代码每条语句的功能：

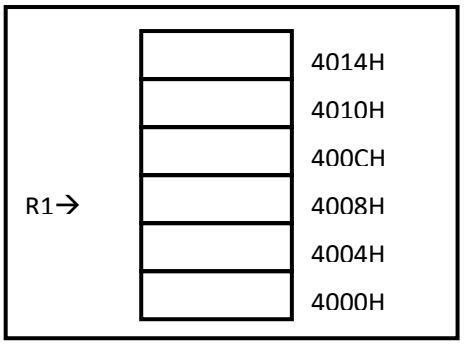
```
R14_<Exception_Mode> = Return Link
SPSR_<Exception_Mode> = CPSR
CPSR[4:0] = Exception Mode Number
CPSR[5] = 0
If <Exception_Mode> == Reset or FIQ then
        CPSR[6] = 1
        CPSR[7] = 1
PC = Exception Vector Address
```

28. 函数定义如下：

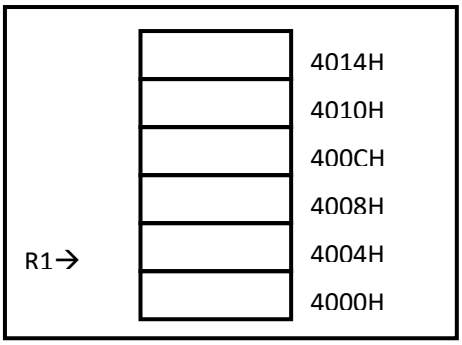
```
void my_strcopy(const char *src,char *dst)
{   int ch;
 __asm {
        loop: LDRB ch,[src],#1;
              STRB ch,[dst],#1;
              CMP ch,#0;
              BNE loop;
 }
}
```

分析程序，说明函数中应用的特色技术及函数功能。

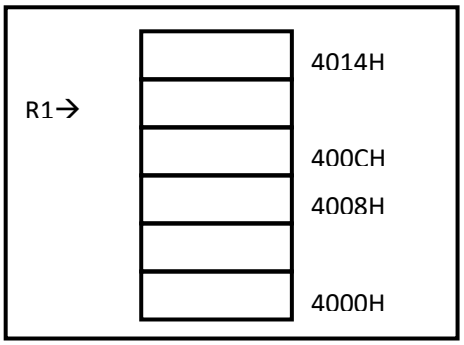
29. 请标出在块拷贝后，寄存器中的值在存储器中对应的位置以及基址寄存器 R1 的变化情况。（执行指令之前基址寄存器为 R1，变址后为 R1’



STMIA R1!,{R3-R5}



STMIB R1!,{R3-R5}



STMDA R1!,{R3-R5}

STMDB R1!,{R3-R5}

30. 下面一段代码是 ARM 代码段跳转到 Thumb 代码段，又回到 ARM 代码段的简单交互程序,请补全其中的代码：

```
        CODE32
        LDR   R0,   =Thumb_Lable+1
        BX    R0

        CODE16
Thumb_Lable   MOV   R1,   #12
…
        LDR   R0,   =Lable
        BX    R0

        CODE32
ARM_Lable   MOV   R1,   #10
```

31. 用 R1 寄存器的最低字节替换掉 R2 寄存器的最低字节，要求不影响条件标志位。
32. 96 位整数减法(被减数从高到低位存放在寄存器 R8、R7、R6 中，减数从高到低位存放在寄存器 R11、R10、R9 中，相减后结果从高到低位存放在寄存器 R5、R4、R3 中)。
33. 用移位指令实现 $A \leftarrow 9A - \frac{B}{2^C}$ (变量 A、B、C 分别存放在寄存器 R1、R2、

R3 中)

34. 用汇编语言实现如下 C 命令（变量 a、b 分别存放在寄存器 R0、R1 中）。

```
C代码：

if(a > b)
      a++;
else
      b++;
```

35. 用汇编语言实现如下 C 命令。（变量 a~e 分别存放在寄存器 R0~R4 中）（3 分）

```
C代码：

if((a = = b)&&(c= = d)) e++
```

36. 在大端模式下将 0x11223344 存储在 0X10000 开始的存储单元中，请写出存储器内容和地址。

37. 堆栈规则为空递减，将 R1 和 R2 的内容入栈和出栈的语句为：

```
STMED    R0!,{R1-R2}
. . .
LDMED    R0!,{R1-R2}
```

试利用数据块传送指令完成以上相同的功能？

38. 下面程序段是取出 SWI 指令中立即数的处理程序，请补全相应代码

```
SWI_Handler
      STMFD    SP!, {R0-R3, R12, LR};
      MRS       R0, SPSR                    ;
      STMFD    SP!, {R0}                   ;
      TST       R0, #0x20              ;
_____;
_____;
_____;
_____;
            ...
LDMFD     SP!, {R0-R3, R12, PC}^    ; SWI 异常中断返回
```

39. 假设寄存器 R0 的内容为 0x40003200，寄存器 R1、R2 的内容分别为 0x11223344 与 0x55667788，采用图示的方法说明下面指令的功能，并解释 ED 的作用。

STMED R0!,{R1,R2}

三、程序设计题

1．数组变量 BUFF 中存放 100 个有符号字数据，编写汇编程序从 BUFF 中找出最大值、最小值，并分别存入变量 MAX、MIN 中。

2．编写汇编程序，实现 1+2+3+…+100 累加功能。

3．编写汇编子程序 add128、sub128 及其测试程序,子程序功能如下，

（1）加法函数 add128

1）入口参数 a 为 128 有符号数；b 为 128 位有符号数。

2）出口参数为 a+b 的结果。

（2）减法函数 sub128

1）入口参数 a 为 128 有符号数；b 为 128 位有符号数。

2）出口参数为 a-b 的结果。

4．编写汇编程序实现将 R1 的高 16 位数据与低 16 位交换，结果保存到地址单元 0x40003000 内。

5．编写汇编程序找出 R0 和 R1 两个数的最大公约数。

6．编写汇编子程序及其测试程序，实现字符串拷贝功能。条件如下，子程序有两个参数，第一个参数是源字符串，第二个参数是目的字符串；字符串以 ASCII 码‘\0’为结束标志。

7．编写汇编程序，实现读取 ARM 指令中 SWI 指令的立即数。

8．编写汇编程序实现 N！功能。

9．编写汇编程序，将十六进制数 0x34AE087C 转化为十六进制数的 ASCII 字符串，结果通过控制台显示。

10．编写汇编程序，完成下列有符号数运算，

（1）0x234A478A+0xBD96AC1F。

（2）0x854B44FA-0x9A43BD32。

（3）0x00868AB2*0x2893A1F0。

（4）0x0A3F7540/0x5。

结果保存到 0x40003100 开始的存储单元中,存储模式为小端模式。

11．编写汇编程序,将内存 0x40003000 开始的连续 600 个字节复制到 0x40003600 开始的地址处。要求,方法一是以 8 个字为单位进行拷贝，不足 8 个字的以字节为单位进行拷贝；方法二是仅以字节为单位进行拷贝。

12．编写字符串比较汇编子程序及其测试程序，比较字符串 str1,str2 的大小，当 str1=str2 时，返回值 0；当 str1>str2 时，返回值 1；当 str1<str2 时，返回值-1。

13．编写汇编程序，实现如下功能，把内存 0x40003000 开始的连续 8 个字节的压缩 BCD 码（内容分别是 0x01，0x23，0x45，0x67，0x89，0x12，0x34，0x56）转换成对应的 ASCII 码，依次存储到 0x40003040 开始的地址处。

14．编写汇编子程序及其测试程序，子程序的功能是将长度为 4 的 ASCII 码数字串转换为等值的二进制数。

15．编写汇编程序，计算 0x40003000 为起始地址连续 14 个字节单元的内容和，如果和小于 100 则将这 14 个字节单元内容复制到 0x40003020 地址处，否则将这 14 个字节单元清零。

16．编写 2 个汇编子程序及其测试程序，一个函数可对给定的整数数组进行从小到大排序；另一个函数可对排序的结果进行折半查找，函数的入口参数、出口参数自定。

17．以 0x40003200 为起始地址连续 100 个字节单元中保存的是 100 个学生的 Java 课程考试成绩，编写汇编语言，统计其中不及格、60-69、70-79、80-89、90-100 的人数，并将统计结果保存在以 0x40003100 为起始地址的连续 5 个字节单元中并通过控制台显示。

18. 某学期，学生考试科目包括英语、数学、C 语言、数据库，编写 C 语言程序实现先从键盘录入考试成绩，然后使用控制台显示学生成绩清单。

19．编写 C 语言程序，将以 0x40003200 为起始地址的连续 1000 个字节单元数据写入文件 MemoryData.dat。

20．编写 C 语言程序，读取文件 MemoryData.dat 的内容，并存储到以 0x40003200 为起始地址的存储单元中。

21.编写汇编子程序实现将存储区 1 的数据拷贝到存储区 2，编程时考虑以下两种情况：

（1）存储区 1 与存储区 2 不重叠。

（2）存储区 1 与存储区 2 重叠，存储区 1 有可能在存储区 2 前，有可能在存储区 2 后。

要求：编写测试程序，测试以上汇编子程序。

22．存储区 1 中的数据已经进行了由小到大排序，编写汇编程序实现将存储区 1 的数据拷贝到存储区 2，拷贝后存储区 2 无重复数据。

其中，存储区 1 的起始地址为 0x40003000，存储大小为 100 字节；存储区 1 的起始地址为 0x40003400。

23．以 0x40003200 为起始地址的连续 100 个字单元中保存的是压缩 BCD 码数据，编写汇编程序，实现将 100 个字单元数据进行 BCD 码加法运算，结果保存到以 0x40003400 为起始地址的单元中。

24．有两个 ASCII 码数据串“115200”、“24”，编写汇编程序实现两数相乘，结果保存到以 0x40003100 为起始地址的单元中。

25.编写汇编程序，统计字数据中 1 的个数，并将统计结果存入字节存储单元 0x40003200 中。

26. 已知 R0=a, R1=b，用汇编语言实现 if ((a!=0x10)&&(b!=0x30)) a=a+b

27、编写汇编程序计算内存 0x40003000 开始的 20 个字节单元数据之和，如果和小于 100 则将这 20 个单元复制到内存 0x40003020 开始的地址处，否则将这 20 个单元清零）

28.汇编编程实现计算表达式 1+2+3+.....+N 的功能.

29.已知 R0=a, R1=b，用汇编语言实现 if (a>b)a++　else　b++

30.用汇编语言实现以下 C 程序功能

```
f=0;
for(i=1;i<=10;i++) {f=f+i}
```

31、数组变量 BUFF 中存放 100 个有符号字数据，编写汇编程序从 BUFF 中找出最大值、最小值，并分别存入字变量 MAX、MIN 中

32. 编写程序将 R1 的高 8 位数据转移到 R0 的低 8 位中，保存到地址 0x40003000 单元内

33. 编写计算 X 的 n 次方的值的汇编程序

34. 有 200 个有符号字节数据，存储在以 0x40003200 为起始地址的内存单元中，试编写汇编程序，找出其中的最大值、最小值以及存储地址，分别存储到内存单元 0x40003101（最大值）、0x40003002（最小值）、0x40003010(最大值地址)、0x40003014(最小值地址)中。

35、编写汇编程序将地址单元 0x40003000～0x40003188 的数据复制到存储单元 0x40003200～0x40003388 中。要求程序中必须使用 LDRB、STRB、LDR、STR、LDM、STM 指令。

36. 用汇编语言实现以下 C 语言语句。

```
For(i=limit;i>=1;i--) {fact=fact*i}
```

37. C 语言程序如下：

```
extern void strcopy(char *d,const char *s)；
…
strcopy(dststr,srcstr)；//该函数实现字符串复制功能，字符串 srcstr 复制到 dststr
…                                    //字符串结束标志是#0
```

编写汇编子程序 strcopy 实现字符串拷贝功能。

38. 利用跳转表的思想编写一个汇编程序，根据键入的值（存放在 R0 中）不同来完成不同的子程序跳转（假设有三个子程序 SUB0、SUB1、SUB2）。

39. 编写一个汇编子程序，完成数据块的复制，源数据区位于 src，目标数据区位于 dst，如下所示。（设堆栈指针初始位于&400）

40.在 ARM 开发工具编译环境下设计程序，用 ARM 汇编语言调用 C 语言实现 20！的阶乘操作，并将 64 位结果保存到寄存器 R0、R1 中，其中 R1 中存放高 32 位结果。

41.先对内存地址 0x4000F000 开始的 100 个字内存单元填入 0x0000001~0x0000064 字数据，然后将每个字单元累加，结果保存于【R9：R8】（R9 中存放高 32 位）。

42.将一个存放在【R1:R0】中的 64 位数据（其中 R1 中存放高 32 位）的高位和低位对称换位，如第 0 位与第 63 位调换，第 1 位与第 62 位调换，第 2 位与第 61 位调换，。。。。第 31 位与第 32 位调换。

43. 编写汇编程序实现 1x2+2x3+3x4+4x5…9x10 表达式的功能

44.用汇编程序实现 C 程序功能

```
char Sendbuf[256];
unsigned char    SendLen;
unsigned char pack(unsigned char CMD,char *buf,unsigned char buflen)
{
 unsigned char i,sum=0;
 if ((buflen<=0)||(buflen>255)) return 0;
 SendLen=0;
 Sendbuf[SendLen++]=0xAA;
 Sendbuf[SendLen++]=CMD;
 Sendbuf[SendLen++]=buflen;
 for (i=0;i<buflen;i++) Sendbuf[SendLen++]=buf[i];
```

```
 for (i=0;i<SendLen;i++) sum+=Sendbuf[i];
 Sendbuf[SendLen++]=sum;
 return 0;
}
```

45.串行通信协议格式如下:

头(Head)	命令(CMD)	数据长度(DataLength)	数据(Data)	校验和(CheckSum)
1Byte	1Byte	2Byte	DataLength Bytes	1Byte
0xAA				前 4 部分字节和

- 发送数据:串口发送数据

数据发送缓冲区　　SendBuf[1024]
数据发送缓冲区长度 SendLen

- 串口接收数据

数据接收缓冲区　　RcvBuf[1024]
数据接收缓冲区长度 RcvLen

编写程序实现下列功能（C 或汇编）:

（1）、将 CMD、发送的数据 DataBuf、发送数据长度 DataLen 按照协议格式打包成可以经过串口直接发送的数据放到 SendBuf 中，其中数据长度放到 SendLen 中。

（2）、经过串口接收的一帧数据保存在 RcvBuf 中，一帧数据的个数保存在 RcvLen 中.编写命令提取函数、校验函数、数据提取函数.