

12.6

```

A:      .word 0xFFFF0010
;
START:  LW     R1, A(R0)
        LW     R2, 0(R1)
        BEQZ   R2, START
        LW     R4, 0(R1)
        J      NEXT_TASK

```

12.7

输入例程:

```

A:      .word 0xFFFF0010
B:      .word 0xFFFF0004
;
START:  LW     R1, A(R0)
        LW     R2, 0(R1)
        ANDI   R3, R2, 2
        BEQZ   R3, START
        LW     R1, B(R0)
        LW     R4, 0(R1)
        J      NEXT_TASK

```

输出例程:

```

A:      .word 0xFFFF0010
B:      .word 0xFFFF000C
;
START:  LW     R1, A(R0)
        LW     R2, 0(R1)
        ANDI   R3, R2, 1
        BEQZ   R3, START
        LW     R1, B(R0)
        SW     0(R1), R4
        J      NEXT_TASK

```

12.8 在屏幕上显示 a 到 z.**13.3** 1). TRAP x0C

2). 可以, 在进入 GETC 与 OUT 两个例程之前, R31 已被储存, 返回后, 程序仍可以按照正常顺序执行.

13.4 1). 每个服务例程占用 $2^8 = 256$ 个存储单元, 256 个共占用 $2^8 * 256 = 2^{16} = 65536$ 个存储单元.

2). 占用 $2^{10} * 256 = 2^{18} = 262144$ 个存储单元.

13.5 程序开始将 x00000018 到 x0000001B 中存入了 0, 造成在执行 TRAP x06 例程时, 程序无法从 TRAP 向量表中找到该例程的起始地址, 导致出错.

13.7 abcd

13.10 1). 不断打印 A, 当任意键被按下时, 有可能输出被按下的字符 (取决于按下时, 程序是否已经执行完 ADDI R4, R0, x41), 并继续输出 A.

2). 程序会不断打印 A, 当 8 被按下时, 有可能输出 8 或者 88 (取决于按下时, 程序是否已经执行完 `ADDI R4, R0, x41`), 并继续输出 A.

13.13 1).

```
printf("(%d)-%d-%d", a, b, c);
```

2).

```
int a, b, c;
scanf("(%d)%d,%d", &a, &b, &c);
```

13.14 1). 0201□

2). 1020CD

3). 001C

14.7 判断 NUM 是否为素数, 如果是, RESULT 为 1, 否则为 0.

14.8 将 DATA 中的数据按照从小到大排序 (冒泡排序).

14.9 R31 的值在进入 SubB 后会改变, 导致无法从 SubA 中返回 main, 可作如下改动:

```
        .data    x30000000
NUM:     .word    #8
+ SaveR31: .space  #4
;

        .text    x40000000
        .global  main
main:    JAL      SubA
        TRAP     x07
        TRAP     x00
;
SubA:    LW       R1, NUM(R0)
+        SW       SaveR31(R0), R31
        JAL      SubB
+        LW       R31, SaveR31(R0)
        RET
;
SubB:    ADDI     R4, R1, x30
        RET
```

14.10 屏幕显示 `Please enter your string:`，用户输入一个字符串，每输入一个字符，屏幕同时回显该字符，直到遇到回车或长度达到 100 结束，随后输出这个字符串的逆序。

11.18

