

考试科目名称 计算机系统基础（上机）

考试方式：开卷 ☒ 闭卷 考试日期 2011 年 1 月 7 日 教师 王浩然
系（专业） 软件工程 年级 班级
学号 姓名 成绩

要求：

1. 在桌面上新建一个文件夹，命名为 exam_101250***（即你的学号）；
2. 两道题目均保存在这个目录下。
3. 交卷时，退出 DLX 模拟器和 VC，即可离开考场。
4. 不允许使用 U 盘、手机等设备。

请仔细审题。

1.(10%) DLX 汇编程序设计：按照字典顺序对比两个字符串，如果二者相同，则 R2=0；如果第一个字符串排在第二个之前，则 R2=-1；否则，R2=1。

要求：

第一个字符串起始于存储单元 x30000000，第二个字符串起始于 x30000020，程序起始于 x40000000。（假设字符串内容已确定，不需要使用 I/O）

2.(10%)编写一个函数 int SearchStr(char *str1, char *str2)，返回字符串 str2 在字符串 str1 中出现的次数。

例如：str1: "abcdefbcbdf", str2: "bcd", 则结果为 2。

main 函数：输入字符串 str1 和 str2（均小于 40 个字符），调用函数 SearchStr 后，输出结果。

大一期末

考试科目名称 计算机系统基础 (A 卷上机)

考试方式: 开卷 ☒ 闭卷 考试日期 ____ 年 ____ 月 ____ 日 教师 王浩然
系 (专业) 软件工程 年级 ____ 班级 ____
学号 ____ 姓名 ____ 成绩 ____

要求:

1. 在桌面上新建一个文件夹, 命名为 exam_121250*** (即你的学号);
2. 两道题目均保存在这个目录下 (该目录下必须包含 *.c 和 *.dlx 文件)。
3. 交卷时, 不要关机, 试卷不要带走, 离开考场即可。
4. 不允许使用 U 盘、手机等设备。

请仔细审题。

1. (10 分) DLX 汇编程序设计: 对一系列数值按照从小到大的顺序排序。 ($n \geq 10$)

要求: 该列数值存储于从 x30000000 起始的存储单元中, 排序后仍位于这些单元中, 程序起始于 x40000000。

2. (10 分) 编写一个函数 int StrFunc (char *str), 将字符串 str (字符串里面包括: 英文字母、空白、负号和数字) 转换为整数。

要求:

- a. 在 main 函数中输入字符串 str (小于 40 个字符), 调用函数 StrFunc 后, 输出结果;
- b. 不能更改函数声明;
- c. 在 StrFunc 函数中不需声明新的数组;
- d. 不能调用 C 标准库中的字符串函数;
- e. 如果字符串不能被转换, 如空串、"abc"、"ab3c"等形式的串, 则返回值为 0; 如果字符串前是空格、TAB、换行等空白, 如" 123ab45", 能进行正确转换 (该例结果应为 123); 如果字符串前是负号, 能进行转换, 如"-123"应被转换成-123。

考试科目名称 计算机系统基础

考试方式: 开卷 ☒ 闭卷 ☐ 考试日期 2013 年 1 月 12 日 教师 王浩然

系(专业) 软件工程 年级 班级

学号 姓名 成绩

题号	1	2	3	4	5	6	7	8	9	10
分数										
题号	11	12	13	14	15	16	17	18	19	20
分数										

得分 1. (10%) 计算: 要求给出关键的计算过程。

(a) 将十进制数 -432 转化为12位二进制补码整数: (2%)

(b) 采用二进制补码整数数据类型, 计算 $1100 + 01111101$, 结果以十进制表示, 并判断是否溢出? (2%)

(c) $((\text{NOT } xB2) \text{ XOR } x37) \text{ AND } (xAC \text{ OR } (\text{NOT } xDF))$, 结果以十六进制表示; (2%)

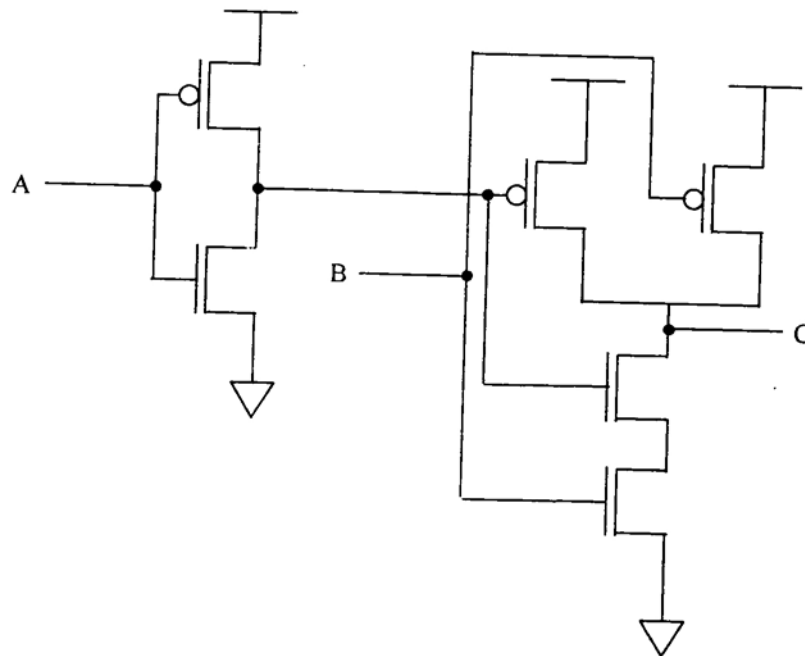
(d) -123.40625的32位二进制编码? (IEEE 754标准) (2%)

ASC II a-97.
A-65

(e) 采用二进制补码整数数据类型，计算 $x789A + x6789$ ，结果以十进制表示，并判断是否溢出？(2%)

得分 2. (10%)

(a) (4%) 给出如下电路图的真值表。



A	B	C

(b) (6%) 使用与、或、非门，给出如下真值表（A、B、C为输入，D为输出）的门级电路图。

A	B	C	D
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

得分 3. (12%)

- (a) 一个有 7 条输入线的译码器，有多少输出线？ (1%)
- (b) 一个有 16 条输入线的多路选择器，有多少输出线？有多少选择线？ (2%)
- (c) 某计算机为 16 位可寻址（即寻址能力），访问一个内存单元也需要 16 位， (3%)
- (i) 该内存的地址空间？
 - (ii) 该内存的大小为多少位？
 - (ii) 该内存的大小为多少字节？
- (d) 冯·诺依曼模型有哪些主要组成部分？ (4%)
- (e) 如果程序在读 KBDR 之前，未检查 KBSR 的就绪位，会发生什么情况？如果程序在写 DDR 之前，未检查 DSR 的就绪位，会发生什么情况？ (2%)

得分 4. (6%)

假设一个机器有如下格式的32位指令：

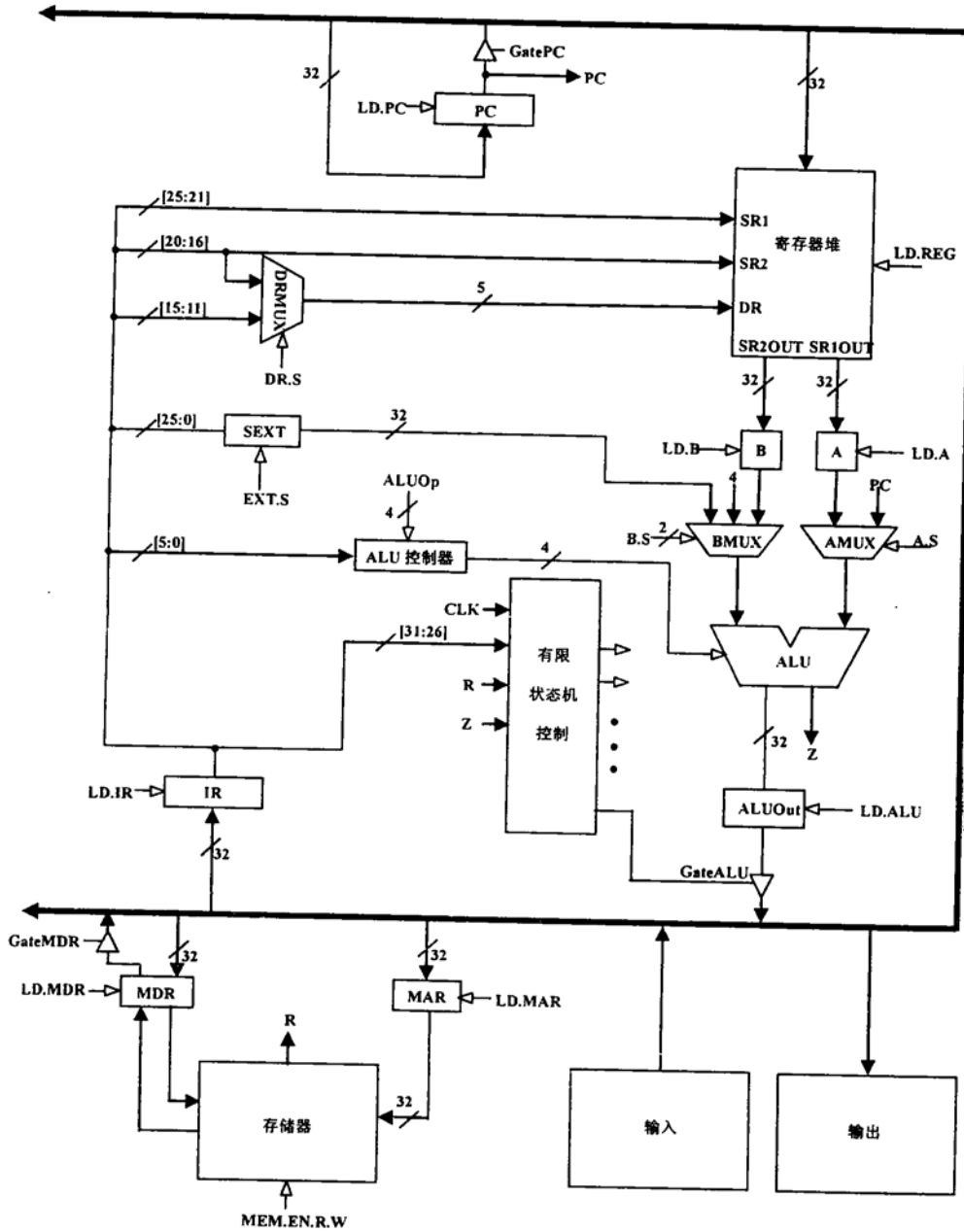
操作码	目标寄存器	源寄存器 1	源寄存器 2	立即数
-----	-------	--------	--------	-----

如果共有 200 个操作码，20 个寄存器，那么，立即数的最大和最小值分别是多少（假设立即数是二进制补码类型）？要求给出关键的计算过程。

得分

5. (5%)

对于指令: SW z(R_x), R_y, 假设该指令的取指令阶段已经完成, 从指令处理的下一阶段开始, 到该指令处理结束, 其数据传输需要经过哪些部分? 请按照指令处理阶段进行描述。



得分	
----	--

6. (6%)下面的程序将计算 $0+1+2+\dots+n$ (n 是从键盘输入的1~9之间的数值), 并将结果存入NUM中。

```
01          .DATA    x30000000
02  NUM:     .SPACE   #4
03  ;
04          .TEXT     x40000000
05          .GLOBAL  MAIN
06  MAIN:    TRAP      x09          ;IN
07          SUBI      R4, R4, x30    ;ASCII->Binary
08          JAL       A
09          TRAP      x00          ;HALT
0A  ;
0B  A:       ADDI      R2, R0, #0
0C          ADDI      R1, R0, #1
0D  LOOP:    BEQZ      R4, EXIT
0E          ADD        R2, R2, R1
0F          ADDI      R1, R1, #1
10          SUBI      R4, R4, #1
11          J         LOOP
12  EXIT:     JAL       B
13          JR        R31
14
15  B:        SW        NUM(R0), R2
16          JR        R31
```

(a) 程序不能正常工作, 为什么? (2%)

(b) 如何修正该错误? (4%)

得分	
----	--

7. (8%)对于如下DLX汇编程序:

```
01          .DATA    x30003000
02  TESTOUT: .ASCIIZ   "abc"
03  ;
04          .TEXT     x40000000
05          .GLOBAL  MAIN
```

```

06  MAIN:    ADDI    R1, R0, TESTOUT
07          ADDI    R2, R0, #0
08  BACK_1:  LB      R4, 0(R1)
09          BEQZ    R4, NEXT_1
0A          TRAP    x07                ;OUT
0B          ADDI    R1, R1, #1
0C          ADDI    R2, R2, #1
0D          J       BACK_1
0E;
0F  NEXT_1:  SUBI    R1, R1, #1
10          LB      R4, 0(R1)
11          SUBI    R2, R2, #1
12          BEQZ    R2, NEXT_2
13          TRAP    #7
14          J       NEXT_1
15;
16  NEXT_2:  TRAP    x00

```

(a) (3%)填写符号表:

标记	地址

(b) (3%)参考下表, 将06、08、09和0D行的指令翻译为机器语言指令。

指令	操作码
LHI	001100
ADDI	000001
LB	010110
BEQZ	101000
J	101100

06:

08:

09:

0D:

(c) (2%)此程序的输出是什么?

得分

8. (8%)在如下程序的空格(a)~(f)处填写指令，使得当程序执行完时，计算出 Fibonacci(NUM)的值（假设 NUM 中存储的数值为 10），并将结果存入 RESULT 中。

```

                .DATA    x30000000
NUM:            .WORD    #10
RESULT:         .SPACE   #4
;
                .TEXT    x40000000
                .GLOBAL  MAIN
MAIN:           _____ (a)
                JAL      A
                _____ (b)
                TRAP     x00
;
A:              BEQZ     R4, EXIT_1
                SLEI     R5, R4, #1
                BNEZ     R5, EXIT_1
                _____ (c)
                ADDI     R1, R0, #1
                ADDI     R2, R0, #1
LOOP:           ADDI     R3, R2, #0
                _____ (d)
                ADDI     R1, R3, #0
                _____ (e)
                SUBI     R4, R4, #1
                J         LOOP
EXIT_1:         ADDI     R2, R0, #1
EXIT:           _____ (f)

```

得分

9. (16%)

如下代码片段的输出各是什么？

(a) (2%)

```

int num = 0;
while (num <= 2){
    num ++;
    printf("%d\n", num);
}

```

输出:

(b) (3%)

```

int i;
for (i = 3; i < 6; i++){
    if (i % 2)
        printf("* * %d", i);
    else
        printf("# # %d\n", i);
}

```

输出:

(c) (2%)

```
int i = 1;
int s = 0;
do {
    s = s + i;
    i++;
} while (i <= 10);
printf("s = %d, i = %d.", s, i);
```

输出:

(e) (2%)

```
char x[20] = "Programming";
char y[20] = "Cprogramming";
int i = 0;
while (x[i] != '\0' && y[i] != '\0'){
    if (x[i] == y[i])
        printf("%c", x[i]);
    i++;
}
```

输出:

(d) (4%)

```
char c[20] = "2473";
int i = 0;
while (c[i] != '\0'){
    switch (c[i] - '2'){
        case 0:
        case 1: putchar(c[i] + 4);
        case 2: putchar(c[i] + 4); break;
        case 3: putchar(c[i] + 3);
        default: putchar(c[i] + 2); break;
    }
    printf("\n");
    i++;
}
```

输出:

(f) (3%)

```
char ch[7] = "12ab56";
int i = 0;
int s = 0;
for (i = 0; ch[i] != '\0'; i = i + 2)
    if (ch[i] >= '0' && ch[i] <= '9')
        s = 10 * s + ch[i] - '0';
printf("%d", s);
```

输出:

得分

10.(8%)下面的程序的功能是将字符串中的字符重新排列:字母从字符串的左端开始向右排列,数字从字符串的右端开始向左排列(如原字符串为“f3bc24n”,重排后为“fbcn423”)。请填空((a)~(f)),将程序补充完整。
#include <stdio.h>

```
void MyFunc (char * str);
```

```
int main() {
    char str[80];
    scanf ("%s", _____ (a));
    MyFunc (str);
    printf ("%s\n", str);
}
```

```
}
```

```
void MyFunc (char * s) {  
    int i, j;  
    int num = 0;  
    int len = 0;  
    char temp;  
    while ( _____ (b))  
        len++;  
    for (i = 0; _____ (c); i++) {  
        if ( _____ (d)){  
            num++;  
            for ( _____ (e); j < len - num; j++){  
                temp = s[j+1];  
                _____ (f)  
                s[j] = temp;  
            }  
            i--;  
        }  
    }  
}
```

得分	
----	--

 11. (11%) C 函数如下:

(a) (2%) 该函数的功能是什么?

(b) (9%) 将如下C函数翻译为DLX汇编代码。(DLX将R4~R7用于参数传递, 将R2和R3用于返回值, 将R16~R23用于局部变量, 将R8~R15和R24、R25用于存储临时产生的值, R29为运行时栈指针。)

```
void MyFunc (char * s) {  
    int i = 0;  
    while (s[i] != '\0'){  
        if (s[i] >= 'a' && s[i] <= 'z'){  
            s[i] = s[i] - ('a' - 'A') + 2;  
            if (s[i] > 'Z' && s[i] <= 'Z' + 2)  
                s[i] = s[i] - ('Z' - 'A' + 1);  
        }  
        i++;  
    }  
}
```

2013.3.5 晚补考卷

考试科目名称 计算系统基础 (B)

考试方式: 开卷 ☒ 闭卷 考试日期 2013 年 月 日 教师 王浩然

系 (专业) 软件工程 年级 班级

学号 姓名 成绩

题号	1	2	3	4	5	6	7	8	9	10
分数										
题号	11	12	13	14	15	16	17	18	19	20
分数										

得分 1. (8%) 计算: 要求给出关键的计算过程。

(a) 将十进制数 -86 转化为8位二进制补码整数; (2%)

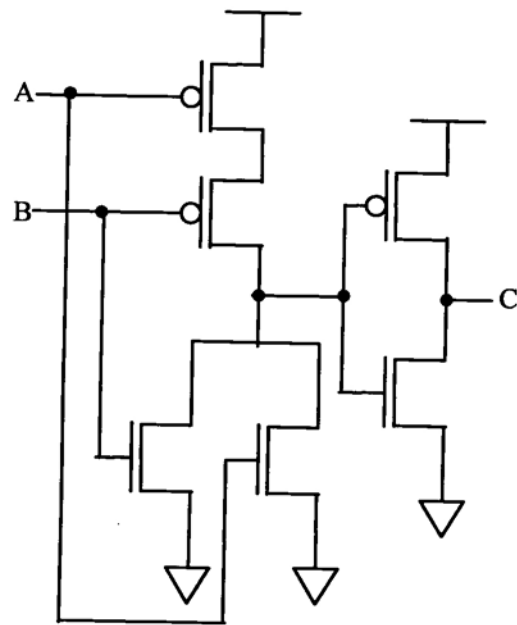
(b) 采用二进制补码整数数据类型, 计算 $1101 + 01010101$, 结果以十进制表示, 并判断是否溢出? (2%)

(c) $((\text{NOT } 1101) \text{ AND } 0101) \text{ OR } (1101 \text{ AND } (\text{NOT } 0101))$, 结果以十六进制表示。 (2%)

(d) 32.9375的32位二进制编码? (IEEE 754标准) (2%)

得分 2. (9%)

(a) (4%) 给出如下电路图的真值表。



A	B	C

(b) (5%) 对于如下真值表，生成门级逻辑电路。

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

得分 3. (12%)

- (a) 一个有16条输入线的多路选择器，有多少输出线？有多少选择线？ (2%)
- (b) 一个有16条输入线的译码器，有多少输出线？ (1%)
- (c) 某计算机为8位可寻址，访问一个内存单元需要16位， (3%)
- (i) 该内存的地址空间？
 - (ii) 该内存的大小为多少位？
 - (ii) 该内存的大小为多少字节？
- (d) 冯·诺依曼模型有哪些主要组成部分？ (4%)
- (e) 请解释轮询I/O与中断驱动I/O的区别。 (2%)

得分 4. (6%)假设一个32位的指令采取如下格式：

操作码	目标寄存器	源寄存器1	源寄存器2	无符号整数
-----	-------	-------	-------	-------

如果共有200个操作码和60个寄存器，“无符号整数”能够表示的最大数是多少？要求给出关键的计算过程。

得分 5. (7%)对于如下程序：

```

        .data      x00006000
NUM:     .word      #100, #5
RESULT:  .space     4
;

        .text      x40000000
        .global    main
main:    addi        r1, r0, NUM
        addi        r4, r0, #0
        lw          r2, 0(r1)
        lw          r3, 4(r1)
LOOP:    beqz        r3, DONE
        add         r4, r4, r2
        subi        r3, r3, #1
        j           LOOP
DONE:    sw          RESULT(r0), r4
        trap        x00

```

1) 构建符号表; (5%)

标记	地址

2) 该程序实现了什么? (2%)

得分 6. (3%)假设在如下程序执行之前, 在 NUM1 中设置了一个正整数的值, 该程序实现了什么?

```

        .data      x30000000
NUM1:    .space     4
NUM2:    .space     4
;

```

```

        .text      x40000000
        .global    main
main:    addi       r1, r0, #0
        lw         r2, NUM1(r0)
        andi       r3, r2, #1
        bnez      r3, NEXT
        subi       r2, r2, #1
NEXT:    add        r1, r1, r2
        subi       r2, r2, #2
        slei       r3, r2, #0
        beqz      r3, NEXT
        sw         NUM2(r0), r1
        trap       x00

```

得分	
----	--

7. (5%)对于如下 DLX 汇编程序:

```

        .data      x30000000
DATA:    .word     #3, #14, #35, #47, #5, #20, #12, #14, #6, #22
SaveR6:  .space    #4
;

        .text      x40000000
        .global    main
main:    addi       r1, r0, DATA
        addi       r2, r0, #9
OutLoop: beqz      r2, EXIT
        addi       r3, r2, #0
InnerLoop: lw       r4, 0(r1)
        lw         r5, 4(r1)
        jal        CMP
        addi       r1, r1, #4
        subi       r3, r3, #1
        bnez      r3, InnerLoop
        slli       r6, r2, #2

```



```

                sub    r1, r1, r6
                subi   r2, r2, #1
                j      OutLoop
EXIT:          trap   x00
;
CMP:          sw      SaveR6(r0), r6
                slt    r6, r4, r5
                bnez   r6, Return
                sw      4(r1), r4
                sw      0(r1), r5
Return:       lw      r6, SaveR6(r0)
                ret

```

(a) (2%)CMP子例程实现了什么？

(b) (3%)该程序实现了什么？

得分 8. (12%)

如下代码片段的输出各是什么？

(a) (3%)

```

int i;
int sum = 0;
for (i = 1; i <= 50; i += 2)
    if (i % 7)
        sum = sum + i;
printf("%d\n", sum);

```

输出:

(b) (3%)

```

int i = 1;
int sum = 0;
do {
    if (i % 7)
        sum = sum + i;
    i++;
} while (sum < 100);
printf("i = %d, sum = %d\n", i, sum);

```

输出:

(c) (3%)

```
int i;
for ( i = 1; i <= 5; i++)
    switch (i % 3) {
        case 0:
            printf("*"); break;
        case 1:
            printf("#"); break;
        default:
            printf("\n");
        case 2:
            printf("&");
    }
```

输出:

(d) (3%)

```
char str[13] = "hello world!";
char *p;
p = str;
while (*p != '\0') {
    printf("%c", *p - 'a' + 'A');
    p++;
}
```

输出:

得分

9. (6%)如下程序的输出是什么?

```
#include <stdio.h>
```

```
int Multiply (int x, int y);
```

```
int z = 4;
```

```
int main () {
```

```
    int x = 2;
```

```
    int y = 3;
```

```
    int z;
```

```
    z = Multiply (x, y);
```

```
    x = Multiply (y, z);
```

```
    y = Multiply (x, z);
```

```
    printf ("%d %d %d \n ", z, x, y);
```

```
}
```

```
int Multiply (int x, int y) {
```

```
    return x * y * z;
```

```
}
```

得分	
----	--

10. (6%) 如下程序的输出是什么?

```
#include <stdio.h>
```

```
void Swap1 (int x, int y);
```

```
void Swap2 (int *x, int *y);
```

```
int main() {
```

```
    int x = 1;
```

```
    int y = 2;
```

```
    printf ("x = %d, y = %d\n", x, y);
```

```
    Swap1 (x, y);
```

```
    printf ("x = %d, y = %d\n", x, y);
```

```
    Swap2 (&x, &y);
```

```
    printf ("x = %d, y = %d\n", x, y);
```

```
}
```

```
void Swap1 (int x, int y) {
```

```
    int temp;
```

```
    temp = x;
```

```
    x = y;
```

```
    y = temp;
```

```
}
```

```
void Swap2 (int *x, int *y) {
```

```
    int temp;
```

```
    temp = *x;
```

```
    *x = *y;
```

```
    *y = temp;
```

```
}
```

得分 11. (4%)编写一个 DLX 程序：判断存储在 R1 中的数是否是偶数。
如果是偶数，则 R2=1，否则，R2=0。

得分 12. (10%)编写 C 程序：计算 π 的近似值。可以使用如下级数展开式来计算。要求项数 n 从键盘输入，并在显示器上输出结果。

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \cdots + (-1)^{n-1} \frac{4}{2n-1} + \cdots$$

得分

13. (12%)编写一个 C 函数: `void StrCat (char *firstStr, char *secondStr)`, 实现字符串连接, 把 `secondStr` 所指的字符串添加到 `firstStr` 所指的字符串的末尾。并使用这个函数写一个程序: 将键盘输入的两个字符串连接后输出。