

第四章 串

一、选择题

1. 下面关于串的的叙述中, 哪一个是不正确的? ()【北方交通大学 2001 一、5 (2分)】

- A. 串是字符的有限序列 B. 空串是由空格构成的串
C. 模式匹配是串的一种重要运算 D. 串既可以采用顺序存储, 也可以采用链式存储

2 若串 $S_1 = 'ABCDEFG'$, $S_2 = '9898'$, $S_3 = '###'$, $S_4 = '012345'$, 执行 $concat(replace(S_1, substr(S_1, length(S_2), length(S_3)), S_3), substr(S_4, index(S_2, '8'), length(S_2)))$

其结果为 ()【北方交通大学 1999 一、5 (25/7分)】

- A. ABC###G0123 B. ABCD###2345 C. ABC###G2345 D. ABC###2345
E. ABC###G1234 F. ABCD###1234 G. ABC###01234

3. 设有两个串 p 和 q , 其中 q 是 p 的子串, 求 q 在 p 中首次出现的位置的算法称为 ()

- A. 求子串 B. 联接 C. 匹配 D. 求串长

【北京邮电大学 2000 二、4 (20/8分)】【西安电子科技大学 1996 一、1 (2分)】

4. 已知串 $S = 'aaab'$, 其 Next 数组值为 ()。【西安电子科技大学 1996 一、7 (2分)】

- A. 0123 B. 1123 C. 1231 D. 1211

5. 串 'ababaaababaa' 的 next 数组为 ()。【中山大学 1999 一、7】

- A. 012345678999 B. 012121111212 C. 011234223456 D. 0123012322345

6. 字符串 'ababaabab' 的 nextval 为 ()

- A. (0, 1, 0, 1, 0, 1, 0, 1, 0, 1)
B. (0, 1, 0, 1, 0, 2, 1, 0, 1)
C. (0, 1, 0, 1, 0, 0, 1, 1)
D. (0, 1, 0, 1, 0, 1, 0, 1, 1)

【北京邮电大学 1999 一、1 (2分)】

7. 模式串 $t = 'abcaabbcbcaabdab'$, 该模式串的 next 数组的值为 (), nextval 数组的值为 ()。

- A. 0 1 1 1 2 2 1 1 1 2 3 4 5 6 7 1 2 B. 0 1 1 1 2 1 2 1 1 2 3 4 5 6 1 1 2
C. 0 1 1 1 0 0 1 3 1 0 1 1 0 0 7 0 1 D. 0 1 1 1 2 2 3 1 1 2 3 4 5 6 7 1 2
E. 0 1 1 0 0 1 1 1 0 1 1 0 0 1 7 0 1 F. 0 1 1 0 2 1 3 1 0 1 1 0 2 1 7 0 1

【北京邮电大学 1998 二、3 (2分)】

8. 若串 $S = 'software'$, 其子串的数目是 ()。【西安电子科技大学 2001 应用 一、2 (2分)】

- A. 8 B. 37 C. 36 D. 9

9. 设 S 为一个长度为 n 的字符串, 其中的字符各不相同, 则 S 中的互异的非平凡子串 (非空且不同于 S 本身) 的个数为 ()。【中科院计算所 1997】

- A. $2n-1$ B. n^2 C. $(n^2/2)+(n/2)$ D. $(n^2/2)+(n/2)-1$ E. $(n^2/2)-(n/2)-1$ F. 其他情况

10. 串的长度是指 ()【北京工商大学 2001 一、6 (3分)】

- A. 串中所含不同字母的个数 B. 串中所含字符的个数
C. 串中所含不同字符的个数 D. 串中所含非空格字符的个数

二、判断题

1. KMP 算法的特点是在模式匹配时指示主串的指针不会变小。()【北京邮电大学 2002 一、4 (1分)】

2. 设模式串的长度为 m , 目标串的长度为 n , 当 $n \approx m$ 且处理只匹配一次的模式时, 朴素的匹配 (即子串定位函数) 算法所花的时间代价可能会更为节省。()【长沙铁道学院 1998 一、1 (1分)】

3. 串是一种数据对象和操作都特殊的线性表。()【大连海事大学 2001 1、L (1分)】

二、填空题

1. 空格串是指 (1)，其长度等于 (2)。【西安电子科技大学 2001 软件 一、4 (2分)】
2. 组成串的数据元素只能是_____。【中山大学 1998 一、5 (1分)】
3. 一个字符串中_____称为该串的子串。【华中理工大学 2000 一、3 (1分)】
4. INDEX ('DATASTRUCTURE', 'STR') =_____。【福州大学 1998 二、4 (2分)】
5. 设正文串长度为 n，模式串长度为 m，则串匹配的 KMP 算法的时间复杂度为_____。
【重庆大学 2000 一、4】
6. 模式串 P= 'abaabcac' 的 next 函数值序列为_____。【西安电子科技大学 2001 软件 一、6 (2分)】
7. 字符串 'ababaaab' 的 nextval 函数值为_____。【北京邮电大学 2001 二、4 (2分)】
8. 设 T 和 P 是两个给定的串，在 T 中寻找等于 P 的子串的过程称为 (1)，又称 P 为 (2)。
【西安电子科技大学 1998 二、5 (16/6分)】
9. 串是一种特殊的线性表，其特殊性表现在 (1)；串的两种最基本的存储方式是 (2)、(3)；
两个串相等的充分必要条件是 (4)。【中国矿业大学 2000 一、3 (4分)】
10. 两个字符串相等的充分必要条件是_____。【西安电子科技大学 1999 软件 一、1 (2分)】
11. 知 U= 'xyxyxyxyxy'; t= 'xxy';
ASSIGN (S, U);
ASSIGN (V, SUBSTR (S, INDEX (s, t), LEN (t) +1));
ASSIGN (m, 'ww')
求 REPLACE (S, V, m) = _____。【东北大学 1997 一、1 (5分)】
12. 实现字符串拷贝的函数 strcpy 为：
void strcpy(char *s, char *t) /*copy t to s*/
{ while (_____)
}
【浙江大学 1999 一、5 (3分)】
13. 下列程序判断字符串 s 是否对称，对称则返回 1，否则返回 0；如 f("abba")返回 1，f("abab")
返回 0；
int f((1)_____)
{int i=0, j=0;
while (s[j]) (2)_____;
for(j--; i<j && s[i]==s[j]; i++, j--);
return((3)_____)
}
【浙江大学 1999 一、6 (3分)】
14. 下列算法实现求采用顺序结构存储的串 s 和串 t 的一个最长公共子串。
程序 (a)
PROCEDURE maxcomstr(VAR s, t : orderstring; VAR index, length : integer);
VAR i, j, k, length1:integer; con:boolean;
BEGIN
index :=0; length :=0; i :=1;
WHILE(i<=s.len) DO
[j:=1;
WHILE (j<=t.len) DO
[IF (s[i]=t[j]) THEN
[k:=1; length1:=1; con:=true;
WHILE con DO
IF (1) THEN [length1:=length1+1;k:=k+1;] ELSE(2)_____;
IF (length1>length) THEN [index:=i; length:=length1;]

```

        (3)____;
    ]
        ELSE (4)____;
    ]
    (5)____;
]
END;
程序(b)
void maxcomstr(orderstring *s,*t; int index, length)
{int i, j, k, length1, con;
  index=0;length=0;i=1;
  while (i<=s.len)
  {j=1;
while(j<=t.len)
{ if (s[i]= =t[j])
{ k=1;length1=1;con=1;
  while(con)
    if (1)____ { length1=length1+1;k=k+1; } else (2)____;
    if (length1>length) { index=i; length=length1;}
    (3)____;
  }
  else (4)____;
  }
  (5)____
} } 【上海大学 2000 一、2 (10分)】

```

15. 完善算法：求 KMP 算法中 next 数组。

```

PROC get _next(t:string, VAR next:ARRAY[1..t.len] OF integer);
BEGIN
  j:=1; k:=(1)____; next[1]:=0;
  WHILE j<t.len DO
  IF k=0 OR t.ch[j]=t.ch[k] THEN BEGIN j:=j+1; k:=k+1; next[j]:=k;END
  ELSE k:=(2)____;
  END;

```

【中山大学 1998 四、1 (4分)】

16. 下面函数 index 用于求 t 是否为 s 的子串，若是返回 t 第一次出现在 s 中的序号(从 1 开始计)，否则返回 0。

例如:s= 'abcdefcdek', t= 'cde' ,则 indse(s, t)=3, index(s, 'aaa')=0 。已知 t, s 的串长分别是 mt, ms

```

FUNC index(s, t, ms, mt);
i:=1; j:=1;
WHILE (i<ms) AND (j<mt) DO
  IF s[i]=t[j] THEN [ (1)____; (2)____ ]
    ELSE [ (3)____; (4)____ ]
  IF j>mt THEN return (5)____; ELSE return (6)____
ENDF;

```

【南京理工大学 1999 三、2 (6分)】

17. 阅读下列程序说明和 pascal 程序, 把应填入其中的 () 处的字句写在答题纸上。

程序说明:

本程序用于判别输入的字符串是否为如下形式的字符串:

W&M\$ 其中, 子字符串 M 是子字符串 W 的字符反向排列, 在此假定 W 不含有字符&和字符\$, 字符&用作 W 与 M 的分隔符, 字符\$用作字符串的输入结束符。

例如, 对输入字符串 ab&ba\$, 11&12\$, ab&dd\$, &\$, 程序将分别输出 Ok. (是), No. (不是)。

程序

```
PROGRAM accept(input,output);
CONST midch=' &' ; endch=' $' ;
VAR an:boolean; ch:char;
PROCEDURE match(VAR answer: boolean);
    VAR ch1,ch2:char; f:boolean;
BEGIN
    read(ch1);
    IF ch1<>endch
        THEN IF (1)
            THEN BEGIN match(f);
                    IF f THEN BEGIN read(ch2); answer:=(2) END ELSE answer:=false
                    END
                    ELSE (3)
            ELSE (4)
        END;
BEGIN
    writeln('Enter String:');
    match(an);
    IF an THEN BEGIN
        (5) IF (6) THEN writeln('Ok.') ELSE writeln('No.')
        END
    ELSE writeln('No.')
```

END. 【上海海运学院 1998 七 (15分)】

18. 试利用下列栈和串的基本操作完成下述填空题。

| | |
|----------------|---------------------|
| initstack(s) | 置 s 为空栈; |
| push(s, x) | 元素 x 入栈; |
| pop(s) | 出栈操作; |
| gettop(s) | 返回栈顶元素; |
| sempty(s) | 判栈空函数; |
| setnull(st) | 置串 st 为空串; |
| length(st) | 返回串 st 的长度; |
| equal(s1, s2) | 判串 s1 和 s2 是否相等的函数; |
| concat(s1, s2) | 返回联接 s1 和 s2 之后的串; |
| sub(s, i, 1) | 返回 s 中第 i 个字符; |
| empty(st) | 判串空函数 |

FUNC invert(pre:string; VAR exp:string):boolean;

{若给定的表达式的前缀式 pre 正确, 本过程求得和它相应的表达式 exp 并返回 "true", 否则 exp 为

空串，并返回“false”。已知原表达式中不包含括弧，opset 为运算符的集合。}

```
VAR s:stack; i,n:integer; succ:boolean; ch: char;
BEGIN
  i:=1; n:=length(pre); succ:=true;
  (1)_; (2)_;
  WHILE (i<n) AND succ DO
    BEGIN ch:=sub (pre, i, 1) ;
    IF (3) THEN (4)
    ELSE IF (5) THEN (6)
    ELSE BEGIN
      exp:=concat ((7)_, (8) );
      exp:=concat ((9)_, (10) );
      (11)_;
    END;
    i:=i+1
  END;
  IF (12) THEN
    BEGIN exp:=concat(exp, sub(pre, n, 1)); invert:=true END
  ELSE BEGIN setnull(exp); invert:=false END
  END;
```

注意：每个空格只填一个语句。【清华大学 1996 八】

四、应用题

1. 名词解释：串 【大连海事 1996 一、10 (1分)】【河海大学 1998 二、5 (3分)】
2. 描述以下概念的区别：空格串与空串。【大连海事大学 1996 三、2、(1) (2分)】
3. 两个字符串 S1 和 S2 的长度分别为 m 和 n。求这两个字符串最大共同子串算法的时间复杂度为 $T(m, n)$ 。估算最优的 $T(m, n)$ ，并简要说明理由。【北京工业大学 1996 一、5 (6分)】
4. 设主串 $S = 'xyxyxyxyxyxyxy'$ ，模式串 $T = 'xyxy'$ 。请问：如何用最少的比较次数找到 T 在 S 中出现的位置？相应的比较次数是多少？【大连海事大学 2001 四 (8分)】
5. KMP 算法(字符串匹配算法)较 Brute(朴素的字符串匹配)算法有哪些改进?【大连海事大学 1996 三、1(2分)】
6. 已知模式串 $t = 'abcaabbabcab'$ 写出用 KMP 法求得的每个字符对应的 next 和 nextval 函数值。【北京邮电大学 1997 三 (10分)】
7. 给出字符串 'abacabaaad' 在 KMP 算法中的 next 和 nextval 数组。【北京邮电大学 2000 三、1 (5分)】
8. 令 $t = 'abcabaa'$ ，求其 next 函数值和 nextval 函数值。【北方交通大学 1994 一 (6分)】
9. 已知字符串 'cddcdececdca'，计算每个字符的 next 和 nextval 函数的值。【南京邮电大学 2000 一 2】
10. 试利用 KMP 算法和改进算法分别求 $p_1 = 'abaabaa'$ 和 $p_2 = 'aabbaab'$ 的 next 函数和 nextval 函数。【东南大学 1999 一、6 (8分)】
11. 已知 KMP 串匹配算法中子串为 babababaa，写出 next 数组改进后的 next 数组信息值(要求写出数组下标起点)。【西南交通大学 2000 二、2】
12. 求模式串 $T = 'abcaabbac'$ 的失败函数 Next(j) 值。【西安交通大学 1996 四、4 (5分)】
13. 字符串的模式匹配 KMP 算法中，失败函数(NEXT)是如何定义的?计算模式串 $p = 'aabaabaaabc'$ 中

各字符的失败函数值。【石油大学 1998 一、2 (10 分)】

14. 设字符串 S= 'aabaabaabaac', P= 'aabaac'

(1) 给出 S 和 P 的 next 值和 nextval 值;

(2) 若 S 作主串, P 作模式串, 试给出利用 BF 算法和 KMP 算法的匹配过程。

【北方交通大学 1998 二 (15 分)】

15. 设目标为 t= 'abcaabbabcabaacbacba', 模式为 p= 'abcabaa'

(1) 计算模式 p 的 nextval 函数值; (5 分)

(2) 不写出算法, 只画出利用 KMP 算法进行模式匹配时每一趟的匹配过程。(5 分)

【清华大学 1998 八 (10 分)】

16. 模式匹配算法是在主串中快速寻找模式的一种有效的方法, 如果设主串的长度为 m, 模式的长度为 n, 则主串中寻找模式的 KMP 算法的时间复杂性是多少? 如果, 某一模式 P=' abcaacabaca', 请给出它的 NEXT 函数值及 NEXT 函数的修正值 NEXTVAL 之值。【上海交通大学 2000 一 (5 分)】

17. 设目标为 S= 'abcaabbcaababababca', 模式为 P= 'babab',

(1) 手工计算模式 P 的 nextval 数组的值; (5 分)

(2) 写出利用求得的 nextval 数组, 按 KMP 算法对目标 S 进行模式匹配的过程。(5 分)

【清华大学 1997 四 (10 分)】

18. 用无回溯的模式匹配法 (KMP 法) 及快速的无回溯的模式匹配法求模式串 T 的 next[j] 值, 填入下面表中:

| | | | | | | |
|---------------------|---|---|---|---|---|---|
| j | 1 | 2 | 3 | 4 | 5 | 6 |
| t | a | a | b | b | a | a |
| kmp 法求得的 next[j] 值 | 7 | | | | | |
| 快速无回溯法求得的 next[j] 值 | b | | | | | |

【北京邮电大学 1992 三、1 (25/4 分)】

19. 在改进了的 (无回溯) 字符串模式匹配中, 要先求 next 数组的值。下面是求 nextval 值的算法。

TYPE SAR=ARRAY[1..m] OF INTEGER;

PTY=ARRAY[1..m] OF CHAR;

PROCEDURE next2(P:PTY;VAR NEXTVAL:SAR);

{在模式 P 中求 nextval 数组的值}

```

1 BEGIN
2 J:=1;NEXTVAL[1]:=0;K:=0
3 REPEAT
4 IF (K=0) OR (P[J]=P[K])
5 THEN [ J:=J+1;K:=K+1;
6 IF P[J]=P[K]
7 THEN NEXTVAL[J]:=NEXTVAL[K]
8 ELSE NEXTVAL[J]:=K ]
9 ELSE K:=NEXTVAL[K]
10 UNTIL J=m
11 END;
```

算法中第 4 行有 P[J]=P[K], 第六行中也有 P[J]=P[K]。两处比较语句相同。请分析说明此两处比较语句的含义是什么? 分析此算法在最坏情况下的时间复杂度是多少? 【北京邮电大学 1993 二、2 (6 分)】

20. 在字符串模式匹配的 KMP 算法中, 求模式的 next 数组值的定义如下:

$$\text{next}[j] = \begin{cases} 0 & \text{当 } j = 1 \text{ 时} \\ \max\{k \mid 1 < k < j \text{ 且 } 'p_1 \dots p_{k-1}' = 'p_{j-k+1} \dots p_{j-1}'\} & \\ 1 & \text{其它情况} \end{cases}$$

请问:

- (1) 当 $j=1$ 时, 为什么要取 $\text{next}[1]=0$?
- (2) 为什么要取 $\max\{K\}$, K 最大是多少?
- (3) 其它情况是什么情况, 为什么取 $\text{next}[j]=1$? 【北京邮电大学 1994 二 (8 分)】

21. 给出 KMP 算法中失败函数 f 的定义, 并说明利用 f 进行串模式匹配的规则, 该算法的技术特点是什么?

【东南大学 1993 一、3 (9 分) 1997 一、2 (8 分) 2001 一、6 (6 分)】

22. 在模式匹配 KMP 算法中所用失败函数 f 的定义中, 为何要求 $p_1 p_2 \dots p_{f(j)}$ 为 $p_1 p_2 \dots p_j$ 两头匹配的真子串? 且为最大真子串? 【东南大学 1996 一、3 (7 分)】

23. 如果两个串含有相等的字符, 能否说它们相等? 【西安电子科技大学 2000 软件 一、3 (5 分)】

24. 设 S_1, S_2 为串, 请给出使 $S_1 // S_2 = S_2 // S_1$ 成立的所有可能的条件 ($//$ 为连接符)。

【长沙铁道学院 1997 三、5 (3 分)】【国防科技大学 1999 一】

25. 已知: $s = '(xyz) *'$, $t = '(x+z) *y'$ 。试利用联结、求子串和置换等基本运算, 将 s 转化为 t 。

【北方交通大学 1996 一、3 (5 分)】【山东科技大学 2002 一、6 (5 分)】

第五部分、算法设计

1. 设 s, t 为两个字符串, 分别放在两个一维数组中, m, n 分别为其长度, 判断 t 是否为 s 的子串。如果是, 输出子串所在位置 (第一个字符), 否则输出 0。(注: 用程序实现)【南京航空航天大学 1997 九 (10 分)】

2. 输入一个字符串, 内有数字和非数字字符, 如: $ak123x456 17960?302gef4563$, 将其中连续的数字作为一个整体, 依次存放到一数组 a 中, 例如 123 放入 $a[0]$, 456 放入 $a[1]$, ...。编程统计其共有多少个整数, 并输出这些数。【上海大学 1998 一 (13 分)】

3. 以顺序存储结构表示串, 设计算法。求串 S 中出现的第一个最长重复子串及其位置并分析算法的时间复杂度。【东南大学 2000 五 (15 分)】

类似本题的另外叙述有:

(1) 如果字符串的一个子串 (其长度大于 1) 的各个字符均相同, 则称之为等值子串。试设计一算法, 输入字符串 S , 以 “!” 作为结束标志。如果串 S 中不存在等值子串, 则输出信息 “无等值子串”, 否则求出 (输出) 一个长度最大的等值子串。

例如: 若 $S = "abc123abc123!"$, 则输出 “无等值子串”; 若 $S = "abceebccadddddaaadd!"$, 则输出 “dddd”。

【华中科技大学 2001】

4. 假设串的存储结构如下所示, 编写算法实现串的置换操作。【清华大学 1995 五 (15 分)】

```
TYPE strtp =RECORD
ch: ARRAY[1..maxlen] OF char;
curlen:0..maxlen
END;
```

5. 函数 $\text{void insert}(\text{char} *s, \text{char} *t, \text{int pos})$ 将字符串 t 插入到字符串 s 中, 插入位置为 pos 。请用 c 语言实现该函数。假设分配给字符串 s 的空间足够让字符串 t 插入。(说明: 不得使用任何库函数)

【北京航空航天大学 2001 六 (10 分)】

6. 设计一个二分检索的算法, 在一组字符串中找出给定的字符串, 假设所有字符串的长度为 4。

- (1) 简述算法的主要思想; (3 分)
- (2) 用 PASCAL 语言分别对算法中用到的类型和变量作出说明; (3 分)
- (3) 用类 PASCAL 语言或自然语言写算法的非递归过程; (8 分)
- (4) 分析该算法的最大检索长度; (3 分)
- (5) 必要处加上中文注释。(3 分)

【山东工业大学 1995 八 (20 分)】

7. 设计一 PASCAL 或 C 语言的函数 atoi(x). 其中 X 为字符串, 由 0--9 十个数字和表示正负数的 ‘-’ 组成, 返回值为整型数值。【浙江大学 1994 二 (7 分)】

8. 已知字符串 S1 中存放一段英文, 写出算法 format(s1, s2, s3, n), 将其按给定的长度 n 格式化成两端对齐的字符串 S2, 其多余的字符送 S3。【首都经贸大学 1998 三、8 (15 分)】

9. 串以静态存储结构存储, 结构如下所述, 试实现串操作 equal 算法。

```
CONST maxlen=串被确认的最大长度
TYPE strtp=RECORD
ch:ARRAY[1..maxlen] OF char;
curlen:0..maxlen
END;
```

(以一维数组存放串值, 并设指示器 curlen 指示当前串长)【北京轻工业大学 1998 一 (12 分)】

10. 编写程序, 统计在输入字符串中各个不同字符出现的频度并将结果存入文件(字符串中的合法字符为 A-Z 这 26 个字母和 0-9 这 10 个数字)。【西北大学 2000 四 (10 分)】

11. 写一个递归算法来实现字符串逆序存储, 要求不另设串存储空间。【西南交通大学 2000 三、2】

12. 已知三个字符串分别为 s=' ab...abcaabcbca...a', s'=' caab', s''=' bcb'。利用所学字符串基本运算的函数得到结果串为: s'''=' caabcbca...aca...a', 要求写出得到上结果串 S''' 所用的函数及执行算法。【东北大学 1998 一、1 (10 分)】

13. S="S1S2...Sn" 是一个长为 N 的字符串, 存放在一个数组中, 编程序将 S 改造之后输出:

- (1) 将 S 的所有第偶数个字符按照其原来的下标从大到小的次序放在 S 的后半部分;
- (2) 将 S 的所有第奇数个字符按照其原来的下标从小到大的次序放在 S 的前半部分;

例如:

S='ABCDEFGHIJKL'

则改造后的 S 为 'ACEGKIJLJHFD'。【中科院计算所 1995】

14. 编一程序, 对输入的一表达式(字符串), 输出其 TOKEN 表示。表达式由变量 A, B, C, 常数(数字) 0, 1, ..., 9, 运算符+, *和括号“(”, “)”组成。首先定义符号的类码:

| | | | | | | |
|--------|--------|--------|---|---|---|---|
| 符 号 | 变 量 | 常 量 | * | + | (|) |
| 类 码 | 0 | 1 | 2 | 3 | 4 | 5 |

其次定义符号的 TOKEN 表示:

| | | |
|-----|------|----------|
| 变量: | 类码 0 | NAMEL 地址 |
| 常量: | 类码 1 | CONSL 地址 |
| * | 类码 2 | |
| + | 类码 3 | |
| (| 类码 4 | |
|) | 类码 5 | |

其中 NAMEL 是变量名表（不允许有相同名），CONST 是常量表（不允许有相同数）。

例如，假设有表达式 $(A+A*2)+2*B*3\#$ ，则将生成如下 TOKENL：

| | | | | | | | |
|----|---|---|---|-----|---|---|---|
| 1: | 4 | | (| 7: | 5 | |) |
| 2: | 0 | 1 | A | 8: | 3 | | + |
| 3: | 3 | | + | 9: | 1 | 1 | 2 |
| 4: | 0 | 1 | A | 10: | 2 | | * |
| 5: | 2 | | * | 11: | 0 | 2 | B |
| 6: | 1 | 1 | 2 | 12: | 2 | | * |
| | | | | 13: | 1 | 2 | 3 |

| |
|---|
| A |
| B |

| |
|---|
| 2 |
| 3 |

【吉林大学 1995 一 (20 分)】

www.vxbus.com