



西安交通大学

程序设计 期末复习讲座

C H E N G X U S H E J I 思源 Q 生活中 M O F U X I J I A N G Z U O

西安交通大学

5.18

XI'AN JIAOTONG UNIVERSITY

钱院学辅·钱学组



主讲人：电气钱2301 马子昂

0. 成绩分布



最高分

100

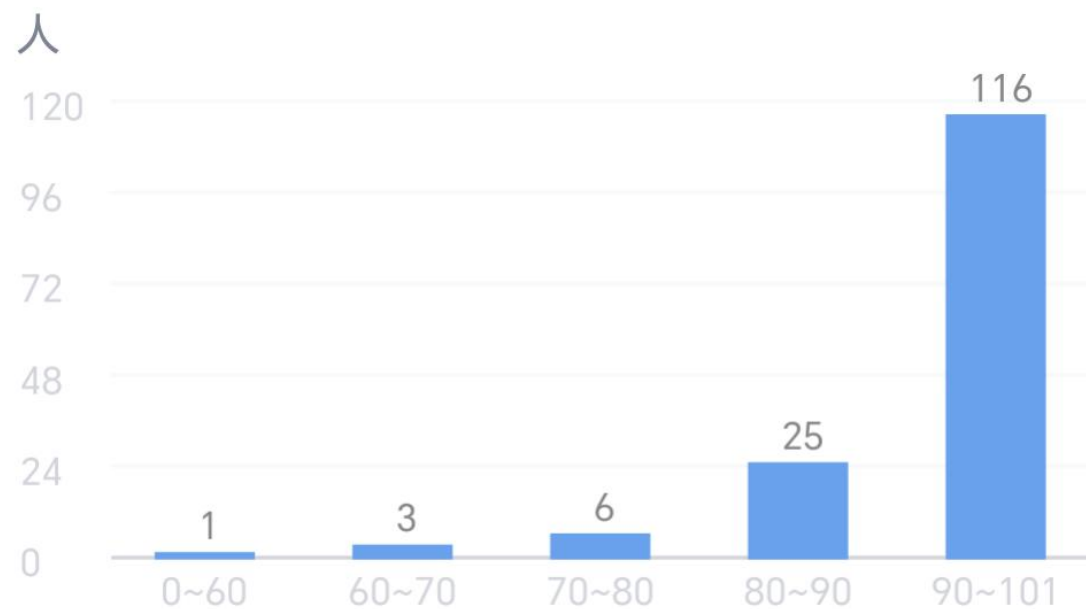
平均分

92.32

最低分

50

成绩分布



22级分布

- 1. 备考目标方法指南
- 2. 知识点回顾
- 3. 真题与例题精讲
- 4. 交流与答疑

01

备考目标方法指南



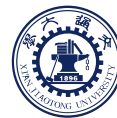
- 1. 试题结构分析
- 2. 备考方向方法
- 3. 考试注意事项

1.1 试题结构分析



- 简答题 $4 \times 8 = 32$ 分（里面可能有填空，23有，22无）；
- 涉及计算和背诵
- 编程5题68分（最高一题20分）；
- 涉及算法和套路实例化

1.2 备考方向方法



- *掌握各部分概念与语法，做到会读会写程序；
- *弄懂弄透复习PPT上的每一道题；
- 动手编写程序，掌握上机作业内容；
- *有一定的手写代码能力（防止考试的时候无处下笔）；
- 养成良好的编程习惯与活跃的编程思维。（不重要）

1.3 考试注意事项



- 注意良好的编程习惯：
 - ✓ 注意命名风格！
 - ✓ 注意缩进与换行规范！
 - ✓ 一定要写注释！
 - ✓ 不要炫技！
 - ✓ 注意细节（尤其是思路性质的细节）！
 - ✓ 少涂改（建议带修正带）！
- 可以考虑在草稿纸上写伪代码或初稿；
- 写不出代码的部分可以写伪代码，起码写思路，不能空题。

02

知识点回顾



- 1. 程序设计基础
- 2. 数据类型&运算符
- 3. 数据输入输出
- 4. 选择结构
- 5. 循环结构
- 6. 数组与字符串
- 7. 函数
- 8. 指针
- 9. 结构体
- 10. 链表
- 11. 类与对象
- 12. 数据结构与算法初步

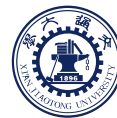
2.1 程序设计基础



➤ 基本概念：

- ✓ 源程序，目标程序，可执行程序
- ✓ 函数，主函数，库函数
- ✓ 机器语言，汇编语言，高级语言
- ✓ 函数，主函数，库函数
- ✓ 常见的库中都有什么函数
- ✓ 全局变量，局部变量
- ✓ 面向对象编程，面向过程编程
- ✓ 递归程序与非递归程序
- ✓ 声明和定义的区别

2.1 程序设计基础 程序=算法+数据结构



- 算法：用系统的方法解决问题的清晰指令
 - ✓ 特性：有穷性，确定性，有零个或多个输入，一个或多个输出，可行性……
 - ✓ 表示方法：自然语言，流程图，N-S流程图，伪代码，计算机语言
 - ✓ 三种基本结构：顺序结构，分支结构，循环结构
- 程序：一组指示计算机每一步动作的指令序列
 - ✓ 与算法的关系？

2.2 数据类型



➤ 算数运算符：

✓ **基本整型**：`int` ($-2147483648 \sim 2147483647$) ；

✓ **短整型** (`short int`)、**长整型** (`long int`)、**双长整型** (`long long int`) ……

➤ 字符型数据 (`char`)：

✓ **大小写字母、数字、一般符号**；

✓ **空格符** (`\t`, `\n`)、**不显示符** (`\0`) 等；

➤ 浮点型数据 (`float`、`double`、`long double`)：可以都用 `float`

➤ 非基本类型：数组、指针、结构体等。

2.2 数据类型-ASCII 码表



ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符
0	NUL	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	X	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93]	125	}

```
char c='2';
```

```
int i=c-48;
```

```
printf("%d",i);
```

2.2 运算符



- 算术运算符 (+ - * / % ++ --)
- 关系运算符 (> < == >= <= !=)
- 逻辑运算符 (! && ||)
- 赋值运算符 (= += -= /= *=)
- 条件运算符 (? :) `grade=score>=90?'A':(score>=60?'B':'C');`
- 指针运算符 (* &)

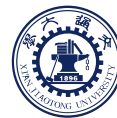
2.2 运算符



➤ 运算符优先次序:

初等运算符	()、[]、->、•
单目运算符	!、++、--、-、(类型)
算术运算符	*、/、%，+、-
关系运算符	<、>、<=、>=，==、!=
逻辑运算符	&&、
条件运算符	? :
赋值运算符	=、+=、-=、*=、/=、%=
逗号运算符	,

2.3 数据输入输出



➤ 数据输出：

`printf("i=%d,c=%c,str=%s,f=%12.4f",i,'\0',"chengxu",15.238531);`//若其中*i*=4，输出是什么？(+右对齐，-左对齐)

答案：i=4, c= , str=chengxu, f= 15. 2385

➤ 数据输入：

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

➤ 单个字符输入输出：`a=getchar()`；`putchar(a)`；

2.4 选择结构



➤ if语句:

```
if (表达式1)    {语句1}  
else if (表达式2) {语句2}  
else if (表达式3) {语句3}  
      ⋮  
else if (表达式m) {语句m}  
else {语句m+1}
```

若if if if呢

➤ switch语句:

```
switch (表达式)  
{ case 常量1: {语句1}break;  
  case 常量2: {语句2}break;  
    ⋮          ⋮          ⋮  
  case 常量n: {语句n}break;  
  default: {语句n+1}  
}
```

break若没有呢

2.5 循环结构



➤ while语句:

while (表达式) {语句}

➤ do-while语句:

do

{语句}

while (表达式);

➤ for语句:

for(初始条件表达式; 循环条件表达式; 循环调整表达式)
{语句}

➤ break & continue

区别?

区别?

2.6 数组与字符串



- 数组: `int a[10]={0, 1, 2, 3, 4};`
- 字符串: `char c[10]=" China" ;char *str1="abcde";` (建议用第一个)

C	h	i	n	a	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----

- 字符串处理函数: (`#include <string.h>`) (不考)

`puts(str), gets(str), strcat(str1, str2), strcpy(str1, str2),
strncpy(str1, str2, n), strcmp(str1, str2), strlen(str),
strlwr(str),strupr(str)...`

2.7 函数



➤ 函数定义：

```
类型名 函数名 (形式参数表列)
{
    函数体
}
```

```
int add (int a, int b)
{
    return (a+b);
}
```

```
int max_value(int array[][4])
{
    int first;
    first = array[0][0];
    return (first);
}
```

➤ 函数声明：int add (int a, int b) ;

➤ 函数的嵌套与递归：例如用递归求n!，实现汉诺塔；

➤ 数组名作为函数参数：

2.7 函数

- 全局变量与局部变量：
- 变量的作用范围：

```
int p=1,q=5;
float f1(int a)
    { int b,c; ..... }
char c1,c2;
char f2 (int x, int y)
    { int i,j; ..... }
int main ( )
{
    int m,n;
    .....
    return 0;
}
```





*唯一的算法

```
void bubble_sort(int arr[], int n) {  
    for (int i = 0; i < n-1; i++) {  
        for (int j = 0; j < n-i-1; j++) {  
            if (arr[j] > arr[j+1]) {  
                // 交换  
                int temp = arr[j];  
                arr[j] = arr[j+1];  
                arr[j+1] = temp;  
            }  
        }  
    }  
}  
  
int main() {  
    int arr[] = {64, 34, 25, 12, 22, 11, 90};  
    int n = sizeof(arr)/sizeof(arr[0]);  
    bubble_sort(arr, n);  
    printf("排序后的数组: ");  
    for (int i = 0; i < n; i++)  
        printf("%d ", arr[i]);  
    return 0;  
}
```


2.8 指针



- 指针即地址，指向变量存储的位置；
- 指针变量可以作为函数参数；
- 通过指针引用（多维）数组： $*(a+n)=a[n]$ ；
- 通过指针引用字符串：`char *string="I love China!"`；
- 指向函数的指针：`int (*p)(int,int)`；`p=max`； //或者 `p=&max`；
- 返回指针值的函数：类型名 *函数名(参数表列)；
- 指针数组：`int *p[4]`；
- 指向指针的指针：`int**p`；
- 内存的动态分配：`malloc`、`calloc`、`free`、`realloc`；

```
int a=1;
int *p; //定义指向int变量的指针
p=&a; //将p指向a
*p=4 //通过指针把a的值改为4

p=(int *)malloc(5*sizeof(int));
```

2.8 指针



- 右左法则：首先从未定义的标识符开始，然后往右看，再往左看。每当遇到圆括号时，就应该掉转阅读方向。一旦解析完圆括号里面所有的东西，就跳出圆括号。重复这个过程直到整个声明解析完毕。

✓ `int (*a)();`

✓ `int (*p)(int *a);`

✓ `char *(*p[10])(int **c);`

✓ `int (*(*p)(int *a))[5];`

2.8 指针



```
char *str1="abcde";  
*(str1++); //可行  
*(str1+2)='f'; //不可行
```

因为本质是 `const char* str1`

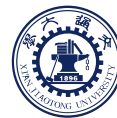
指针常量: `int* const p`
指针类型 常量 指针名

本质是一个常量，而用指针修饰它。
指针指向的地址不可以修改。
指针指向地址的内容可以改变。

常量指针: `const int* p`
常量 指针 (指针类型 指针名)

常量指针本质是指针，并且这个指针是一个指向常量的指针，指针指向的地址可以修改；
指针指向地址的内容不可以改变（不可以通过指针赋值）；

2.8 指针



内存四区的划分包括 代码区、全局/静态区、栈区 和 堆区。

1. 代码区

用于存储程序的机器指令。

包括程序的可执行代码（如函数体的汇编指令）。

只读，存储内容在程序启动时加载到内存中，一直保留到程序结束。

不允许程序员修改其内容

2. 全局/静态区：

存储全局变量、静态变量、常量等。

系统管理，程序员无需手动释放。

3. 栈区：

用于存储局部变量、函数参数、返回地址等。

局部变量的内存分配和释放由编译器自动完成。

当函数调用结束时，栈帧会被弹出，局部变量的内存被释放。

4. 堆区：

用于存储动态分配的内存。

由程序员手动分配和释放内存。

如果未释放分配的内存，就会发生内存泄漏。

new 和 delete (C++) 或 malloc 和 free (C)

2.9 结构体



➤ 结构体：

```
struct 结构体名  
    { 成员表列 };
```

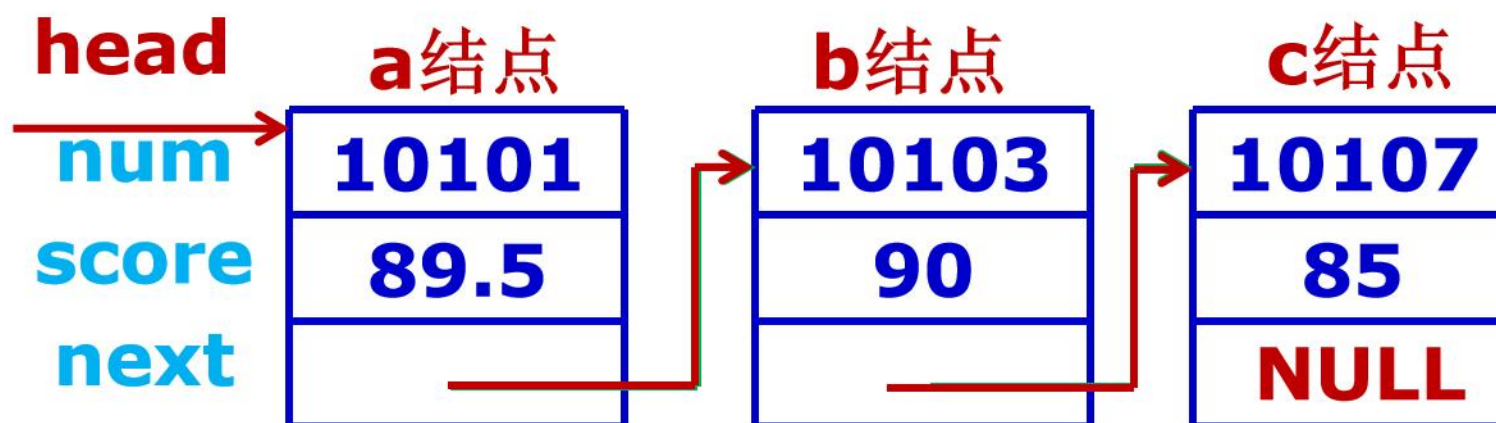
```
struct Node  
{  int num;  
    char name[20];  
    float score;  
    struct Node *next;  
};
```

```
struct Node a,b;  
a.num=1001;  
b.score=70.5;  
struct Node *p;  
p=&a;  
p->next=&b;  
p->next->score=71.5;
```


2.10 链表

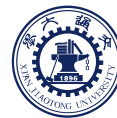


➤ 静态链表:



➤ 动态链表:

开辟节点 `p=(struct Node*)malloc(LEN);`



2.11 类与对象

- 类的定义（属性+方法）：

```
class 类名
{
    成员访问权限:
    成员变量定义;
    成员访问权限:
    成员函数定义;
    .....
};
```

- 内联函数；
- 构造函数、析构函数；
- 多文件结构；

- 成员变量：

数据类型 成员变量名；

- 成员函数：

✓ 在类内定义：

返回类型 成员函数（数据类型 形式参数1）

{

函数体

}

✓ 在类外定义：

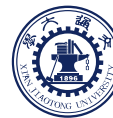
返回类型 类名::成员函数（数据类型 形式参数1 ,.....）

{

函数体

}

2.11 类与对象 (C++)



➤ 对象定义与调用：

关键字
class classname ← 类名

{

Access specifiers: ← // 访问修饰符: private/public/protected

 Data members/variables; // 变量

 Member functions() {} // 方法

}; ← // 分号结束一个类

➤ 函数声明定义方式

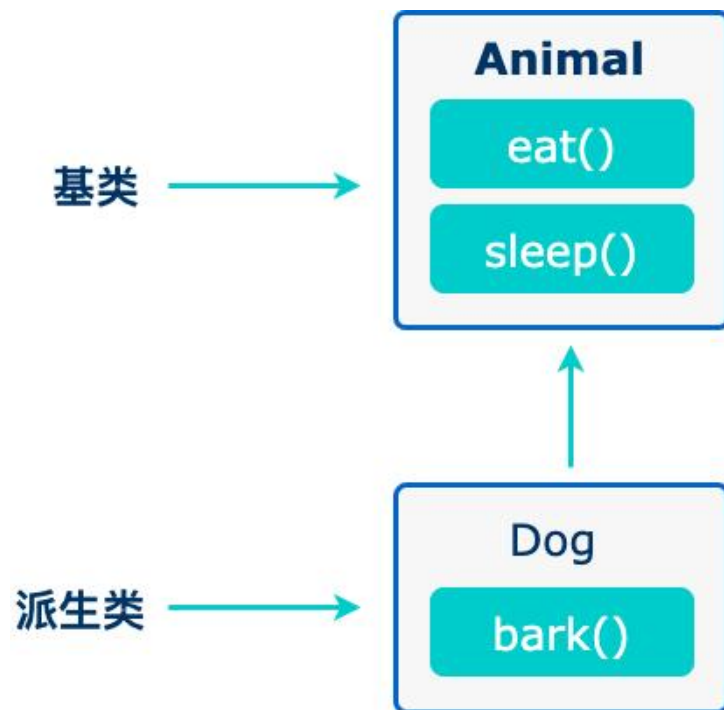
```
class Box
{
    public:
        double length;    // 长度
        double breadth;   // 宽度
        double height;    // 高度
        // 成员函数声明
        double get(void);
        void set( double len, double bre, double hei );
};
// 成员函数定义
double Box::get(void)
{
    return length * breadth * height;
}

void Box::set( double len, double bre, double hei)
{
    length = len;
    breadth = bre;
    height = hei;
}
```

2.11 类与对象 (C++)



➤ 继承



```
// 基类
class Animal {
    // eat() 函数
    // sleep() 函数
};

//派生类
class Dog : public Animal {
    // bark() 函数
};
```

2.12 数据结构与算法初步

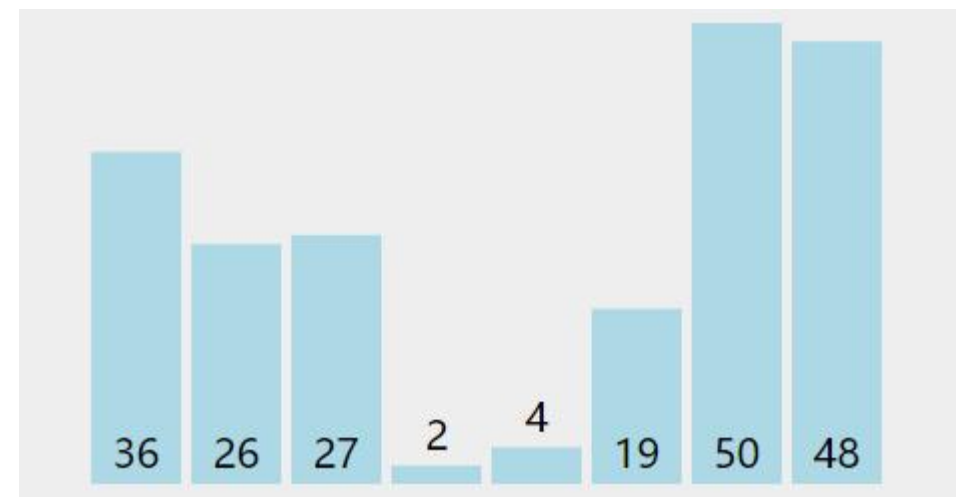


- 排序算法：插入排序、**冒泡排序**、选择排序法；
- 受限线性表：**栈**与队列；
- 特殊的链表：双向链表、循环链表；
- 非线性结构：树与图。
- 贪心算法、分治算法等；

2.12 冒泡排序



```
void bubble_sort(int arr[], int len) {  
    int i, j, temp;  
    for (i = 0; i < len - 1; i++)  
        for (j = 0; j < len - 1 - i; j++)  
            if (arr[j] > arr[j + 1]) {  
                temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
}
```



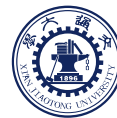
03

真题与例题精讲



西安交通大学
XI'AN JIAOTONG UNIVERSITY

3.1 去年题



1. (☆☆☆) `int a[3][4] = {1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233};`

设数组的首地址为2100，一个int数据占4字节，则：

a为_____

*a为_____

a+2为_____

*a+2为_____

*(a+1)+2为_____

**a为_____

*(a+9)为_____

(a+1)[1]为_____

3.1 去年真题



西安交通大学
XI'AN JIAOTONG UNIVERSITY

简答题

简述C++的优点和特点

简述冯诺依曼架构

3.1 去年真题



编程题

1. 输出100以内的所有质数
2. 冒泡排序一个数组
3. 类的创建和实例化
4. 给一个表头指针进行链表的某一值删除

3.1 前年真题



一、简答题

```
1、 switch(A){  
    case A :  
        printf("This " );  
    case B:  
        printf("is " );  
    case C:  
        printf("a " );  
        break;  
    case D:  
        printf("statement\n");  
}
```

3.1 前年真题

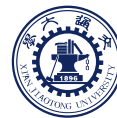


一、简答题

2. 阅读下面代码，写出程序输出。

```
#include<stdio.h>
void main()
{
    int number = 21;
    while(number>0)
    {
        printf("%d\t",number++%3);
        number/=3;
    }
}
```

3.1 前年真题



一、简答题

3. 解释下面各式中p的意义。

`int *p();`

`int *p[4];`

`int (*p)();`

`int (*p[4])();`

3.1 前年真题



一、简答题

4. 解释全局变量与局部变量的区别。

5. 解释递归函数与非递归函数的区别。

3.1 前年真题



二、“水仙花数”是指一个三位数，其各位数字的立方和恰好等于该数本身，如： $153=1^3+5^3+3^3$ ，则153是一个“水仙花数”。编写程序，输出100~999间的所有水仙花数。（12分）

3.1 前年真题



西安交通大学
XI'AN JIAOTONG UNIVERSITY

三、求一元二次方程的根。（12分）

3.1 前年真题



西安交通大学
XI'AN JIAOTONG UNIVERSITY

四、完成一个学生类，类中含有函数，实现设置学生信息、输出学生信息。（12分）

3.1 前年真题

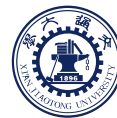


五、给定一个链表（结点结构如下），每个结点包含一个整形数(num)，假设头节点包含数据，已给定指向头结点的指针为(struct Node*) head。该链表已经按照每个节点的num从大到小排序。（20分）

```
struct Node
{
    int num;
    struct Node *next;
};
```

- (1) 将此链表改为从小到大排序。
- (2) 按大小顺序插入新节点，若已有节点则不插入。

3.1 前年真题



西安交通大学
XI'AN JIAOTONG UNIVERSITY

六、使用递归方法，求斐波那契数列的第 n 项值， n 从0开始。
(改编，12分)

3.2 练习题



- 2、(☆☆) 编写程序，输出1-100中所有的素数。
- 3、(☆) 给定一个数组，编写程序将里面所有的数据按原顺序
逆序排列并重新存入这个数组。

3.2 练习题



4、(☆☆☆☆) 使用递归编写程序，使之能够输出 $n=5$ 时汉诺塔所有圆片从A柱移至C柱的逐步方法（例如：逐行输出“从A移至C” “从A移至B” ……）。

5、(☆☆) 编写程序使之能够读取输入的十个数字，使用“冒泡法”排序并输出。（输入输出格式自定）

6、(☆☆☆) 使用指针，写一函数实现两个字符串的比较。即写一个strcmp函数，定义开头如下：

```
int strcmp(char *p1, char *p2)
```

3.2 练习题



7、(☆☆☆☆) 使用指针（不使用链表）编写程序解决以下问题：有n个人（编号1至n）围成一圈顺序报号。从第1个人开始报数（从1至3报数），凡报到3的退出圈子，求最后留下的人的初始编号。

8、(☆☆☆☆☆) 已有a、b两个链表，每个链表中的结点包括学号、成绩。编写程序将两个链表合并为一个，按学号升序排列。两个链表的结点结构、头指针自行定义。

9、(☆☆☆☆) 使用链表解决第7题。

3.2 练习题

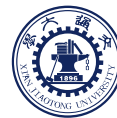


10、(☆☆☆) 给定矩阵 $A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 2 & 4 & 1 \\ 3 & 1 & 2 & 3 \end{bmatrix}$, 使用指针编写程序统计A中有多少个子矩阵（含 1×1 和自身）所有元素的和小于10。

11、(☆☆) 简述下面convert函数的功能。

```
1 char digits[] = { '0', '1', '2', '3', '4', '5', '6', '7'
2   , '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' }; //全局变量
3 void convert(int y, int x){ //x、y都是一定范围内的整数
4     if(x != 0){
5         convert(y, x / y);
6         printf("%c", digits[x % y]);
7     }
8 }
9
```


3.2 练习题



- 12、(☆☆) 简述C语言中指针的几种用法。
- 13、(☆☆) 分别说明在哪些情况下程序开头需要引入stdio.h、string.h、math.h? 如果未引入会发生什么?
- 14、(☆☆) 函数的声明与定义有何异同?
- 15、(☆☆) while循环与do-while循环有何异同?

04

交流与答疑



4.1 群二维码



4.2 学习网站



西安交通大学
XI'AN JIAOTONG UNIVERSITY

<https://www.runoob.com>

MOOC浙大翁恺

聆听 感谢

西安交通大学



XI'AN JIAOTONG UNIVERSITY



西安交通大学