



C程序设计语言

复习指导




第1章 C语言概述与算法

- C语言的基本特点
- C语言程序的结构
- 常用的算法表示方法
- 结构化程序设计方法的基本含义



第2章 数据类型、运算符和表达式

- C语言的标识符组成规则
- 常量的定义方法
- 变量的定义方法
- 基本数据类型：表示方法，取值范围，分类及所占存储空间大小，所适用的运算及运算方法



第2章 数据类型、运算符和表达式

- 几种基本运算（算术运算、关系运算、逻辑运算、赋值运算、逗号运算、强制转换运算）的运算方法，**混合运算规则、运算的级别**；
- 几种表达式（算术、关系、逻辑、赋值、逗号、强制转换表达式）的书写方法及求值方法；
- 特别注意强制转换运算、自增自减运算、赋值运算、复合赋值运算、逗号运算以及逻辑运算的运算方法及相应表达式的求值方法



第3章 顺序结构程序设计

- C语言语句的分类，注意复合语句和空语句
- 赋值语句的执行机理；
- C语言的输出函数printf, putchar的使用方法及格式输出特点
- C语言的输入函数scanf, getchar的使用方法及格式输入特点



第4章 选择结构程序设计

- **if**语句、**if-else**语句、嵌套**if**语句与**if-else**语句：注意逻辑表达式的写法；注意复合语句的使用方法；注意**else**与**if**配对的方法；
- 条件运算与条件表达式：三目运算的运算方法、运算级别以及条件表达式的求值方法；
- **switch**语句：格式，执行方式，与**break**语句的结合；与嵌套的**if-else**语句的不同；



第5章 循环结构程序设计

- **While**语句与**do-while**语句的执行方式，异同比较、适用场合；
- **For**语句的基本写法、执行方式、变异形式及与**While**语句、**do-while**语句的联系；
- **Break**语句和**continue**语句：作用，它们的不同；
- 几种基本题型：累加求和、连乘求积、有规律图形的打印输出（特别注意变量赋初值的位置和初始值）



第6章 函数

- 定义函数的方法：注意函数类型和形式参数类型
- 调用函数的方法：
 - 形式参数与实在参数个数要一样，类型要一致；
 - 参数传递遵循“值传递”原则；
 - 调用方式分为表达式调用和语句调用；
 - 调用时遵循“先定义后使用”的原则；要使用后面定义的函数，可以通过声明函数原型的方法；
 - 理解调用函数的调用机理（计算实在参数表达式——为局部变量分配内存空间——执行函数体语句——返回函数值——释放已经分配的内）；



第6章 函数

- 函数的递归调用：
 - 定义递归函数要注意在递归函数内设置“结束递归的语句”；
 - 掌握递归函数的特点——生成“栈”，利用栈的特点——“先进后出”解决一些要倒序输出的问题；
 - 注意可采用递归技术解决的问题的特点形式及转换方法；
 - 递归的特点是编程简单，但往往效率较递推低，所以要合理使用递归技术；



第6章 函数

- 局部变量与全局变量：使用两种变量的优缺点比较；使用局部变量和全局变量的一般原则；作用域的确定方法；
- 变量的存储类别：特别注意**extern**类别和**static**类别的使用方法与作用；
- 外部函数与内部函数：以**extern**类别和**static**类别标明，它们与一般函数的不同是作用域不同



第6章 函数

- 函数与指针：指针变量作形式参数与非指针变量作形式参数的不同；指针值函数的声明方法及使用方法；指向函数的指针变量：类别（*指针变量名）（形式参数）；
- 了解多文件程序的组织方法：包含文件方法和工程文件方法；
- 了解函数在结构化程序设计中的作用。



第7章 数组

- 一维数组
 - 定义及初始化方法；
 - 常用操作——引用、遍历、排序删除元素、插入元素；
 - 数组的地址与每个元素的地址；
 - 用指向数组的指针变量实现对数组进行操作的方法；



第7章 数组

- 二维数组
 - 定义及初始化方法；
 - 常用操作——引用、遍历、排序删除元素、插入元素；
 - 数组的地址与每个元素的地址；
 - 用指向数组的指针变量实现对数组进行操作的方法；



第7章 数组

- 字符串：
 - 定义与初始化方法
 - 输入输出方法
 - 与字符数组的不同
 - 常用的字符串处理函数
 - 字符串数组
- 数组作函数的参数的特殊性
- 指针数组



第8章 结构体、共用体、枚举

- 主要掌握结构体的定义、初始化、引用成员的方法
- 成员运算符与指针运算符的级别比较；
- 结构体数组的应用
- 共用体变量的定义方法、使用方法
- 共用体与结构体的不同
- 枚举变量的定义、使用方法




第9章 动态数据结构

- 了解动态数据结构的特点
- 掌握内参申请函数的使用方法
- 掌握链表的建立与使用，链表的常见操作
- 了解二叉树的建立方法、操作方法及应用




第10章 文件

- C语言文件的分类及特点
 - 文本文件
 - 二进制文件
- 常见的文件操作
 - 建立、打开；
 - 读、写



习题——判断题

1. C语言的运算不仅有不同的级别，还分为左结合、右结合两种结合方式； 【 】
2. 逗号表达式的值是包含于其中的最后一个表达式的值； 【 】
3. 通过指针形式参数可以弥补函数只能返回一个值的不足； 【 】
4. for语句中位于for后面括号中的3个表达式都可以省略； 【 】
5. 预处理命令是C语言的组成部分； 【 】
6. 局部变量随着所在的函数的调用而诞生，随着调用结束而失效； 【 】



习题——判断题

7. C语言的函数调用时，参数传递遵循“值传送”原则；☐
8. 函数内的局部变量将屏蔽掉与其重名的全局变量；☐
9. 共用体变量与其所有成员共享同一个内存地址；☐
10. C语言程序都是从main()函数开始执行；☐
11. 全局变量的作用域是所在的整个源程序；☐
12. if—else语句中的else总是与离它最近的if配对；☐




习题——判断题

- 13. 用户无法改变递归过程中生成的“栈”中的数据
的访问次序； ☐
- 14. 指针变量和指针所指向的变量是相同的变量；
☐
- 15. C语言对文本文件和二进制文件均采用“缓冲文
件系统”； ☐
- 16. static型局部变量的作用域和全局变量的作用域
一样； ☐
- 17. 多种类型变量组成的表达式求值时进行类型的
自动转换； ☐
- 18. 数组指针和指针数组是一个意思；
☐




习题——判断题

- 19. goto语句的使用不符合结构化程序设计思想;
【 】
- 20. 位运算可以实施对整型变量位的操作;
【 】
- 21. 二维数组初始化时列规模可以省略不写;
【 】
- 22. 删除、插入操作简单高效是动态数据结构相对于静态数据结构的优点之一 【 】
- 23. 数组名作函数参数, 实参数组和形参数组大小可不同, 但类型须相同 【 】
- 24. 静态局部变量的初始化只在第一次调用时进行, 其他次调用采用新值 【 】




习题——判断题

25. 通常把产生内存分配的声明叫做定义 **【 】**
26. `continue`语句和`break`语句都可以用于改变`switch`语句的执行方式 **【 】**
27. `do-while`语句和`while`语句在任何相同情况下，执行效果都一样 **【 】**




习题——选择题

- 可作为变量名的字符序列是 【 】
A) static B) INT C) #ptr_1t D) 123
- C语言中各种基本数据类型的存储空间长度顺序是 【 】
A) $\text{char} \leq \text{long} \leq \text{int} \leq \text{float} \leq \text{double}$
B) $\text{double} \leq \text{float} \leq \text{long} \leq \text{int} \leq \text{char}$
C) $\text{char} \leq \text{int} \leq \text{long} \leq \text{float} \leq \text{double}$
D) $\text{float} \leq \text{int} \leq \text{long} \leq \text{char} \leq \text{double}$



习题——选择题

- 若变量已正确定义，要将a 和b中的数进行交换，不正确的语句组是【 】
A) $a=a+b, b=a-b, a=a-b;$ B) $t=a; a=b; b=t;$
C) $a=t; t=b; b=a;$ D) $t=b; b=a; a=t;$
- 能表示“能被2整除，且能被3或5整除的整数x”的表达式是【 】
A) $(x\%2==0)\&\& (x\%3==0)\|(x\%5==0)$
B) $x\%2==0\&\&x\%3==0\| x\%5==0$
C) $(x\%2==0\&\& x\%3==0)\|x\%5==0$
D) $x\%2==0\&\&(x\%3==0\| x\%5==0)$




习题——选择题

- 以下由**while**构成的循环执行的次数是【 】
A) 无限次 B) 有语法错，不能执行
C) 一次也不执行 D) 执行1次

```
int k = 0;            while ( k = 1 ) k ++ ;
```

- 是字符串常量的是【 】
A) 'a' B) "a" C) 'abc' D) abc



习题——选择题

- 若有以下定义：

```
int a[] = { 1,2,3,4,5,6,7 };c
```

```
char c1 = 'b', c2 = '2';
```

则值不为2的表达式是 【 】

A) a[1]

B) 'd'-c1

C) a['3'-c2]

D) c2-0

- 表达式 'a'-0 的值是 【 】

A) 'a'

B) "a"


C) 字符 'a' 的序号

D) 不确定



习题——选择题

- 已知 `int *p,a;` 则语句 `p=&a;` 中 ‘&’ 的含义是【 】
 - A) 与运算
 - B) 取指针内容
 - C) 赋值运算
 - D) 取变量地址
- 与表达式 `i+1+1` 值相等的是【 】
 - A) `(i++)+1`
 - B) `++i+1`
 - C) `(i++)++`
 - D) `i+1++`



习题——选择题

- 已知 `char *p,*q`;则下列正确的语句是【 】
 - A) `p*=3`;
 - B) `p/=q`;
 - C) `p+=3`;
 - D) `p+=q`;
- 运算符有优先级，除运算符！外，正确的叙述是【 】
 - A) 逻辑运算符高于算术运算符，
算术运算符高于关系运算符
 - B) 算术运算符高于关系运算符，
关系运算符高于逻辑运算符
 - C) 算术运算符高于逻辑运算符，
逻辑运算符高于关系运算符
 - D) 关系运算符高于逻辑运算符，
算术运算符低于逻辑运算符



习题——选择题

- 与表达式 $a-=b-(c+2)$ 等价的是【 】
 - A) $a=a-b-(c+2)$
 - B) $a=a-b+(c+2)$
 - C) $a-b-(c+2)$
 - D) $(a-=b)-(c+2)$
- 对于语句`scanf("%d,%c,%d",&k,&c,&d);`使变量`k,d`(int型), `c`(char型)的值分别为2, 35, 'a'的输入为(□代表一个空格)【 】
 - A) 2□a□35
 - B) 2a35
 - B) C) 2,a,35
 - D) 2□a35



习题——选择题

- 设有说明：char k, m, *p=&m, a[10]; 则正确的调用函数scanf的实在参数为【 】
 - A) (“%c%c%s”, &k, &p, &a)
 - B) (“%c%c%s”, k, p, a)
 - C) (“%c%c%s”, &k, p, a)
 - D) (“%c%c%c”, &k, p, a[1])
- 设有程序段：char c='a'; switch(c)
{default: putchar(c++); case 'c':
putchar(c++); case 'd':putchar(++c); };
则执行结果为【 】
 - A) a B) b C) abc D) abd




习题——选择题

- 以下正确的描述是【 】
 - A) `continue` 语句的作用是终止整个循环的执行。
 - B) `continue` 语句的作用是结束本次循环的执行。
 - C) `continue` 语句和`break` 语句都可以在`switch`语句中使用。
 - D) `break` 语句的作用是结束本次循环的执行。
- 设 $a=13, b=4, c=5$, 则 $!(a+b)+c-1 \&\& b+c/2$ 的值为【 】
 - A) -1
 - B) 0
 - C) 1
 - D) 2



习题——选择题

- 与语句 “ while(!x)”等价的语句是
A) while(x==0) B) while(x!=0)
C) while(x!=1) D) while(x)
- 所包含的均是合法的用户标识符的选项是【 】
A) b-b abc P#d
B) _isw ssiped INT
C) hiy <fr> max
D) float CCP void



习题——选择题

- 已知char str1, str2[10]; 正确的输入语句是 【 】
 - A) scanf ("%c%s", &str1, str2);
 - B) scanf ("%s%s", &str1, str2);
 - C) scanf ("%c%c", &str1, str2);
 - D) scanf ("%s%c", &str1, str2);
- 若希望当A的值为奇数时,表达式的值为"真"; A的值为偶数时, 表达式的值为"假"。不能满足要求的表达式是 【 】
 - A) A%2==1
 - B) !(A%2== 0)
 - C) ! (A%2)
 - D) A%2




习题——选择题

- 设有程序段： `char c='a'; switch(c) {default: putchar(++c); break; case 'c': putchar(c++); case 'd':putchar(++c); }`； 则执行结果为 【 】
A) a B) b C) c D) d
- 设有函数： `sub(int k){if(k%10==k*k%10) sub(k+1); printf("%d,"k);}` 则调用`sub(12)`的执行结果为 【 】
A) 12, 13, 14, 15, B) 15,
C) 15, 14, 13, 12, D) 14, 13, 12,



习题——选择题

- 设有说明： `struct AA{ char a, *b, c[10]} s;` 调用函数 `scanf` 的正确的实在参数为 【 】
 - A) (“%c%c%s”, &s.a &s.b, &s.c)
 - B) (“%c%c%s”, s.a, s.b, s.c)
 - C) (“%c%c%s”, &s.a, s.b, s.c)
 - D) (“%c%c%c”, &s.a, s.b, s.c[1])
- 设有说明： `struct AA{ int a,b} *s;` 调用函数 `scanf` 的正确的实在参数为 【 】
 - A) (“%d”, &*s.a) B) (“%d”, *s.a)
 - C) (“%d”, &s->a) D) (“%d”, s->a)




习题——选择题

- 程序段 `int k=0; while(1) {k++; if(k==5) break; }` 中循环体执行的次数是 **【 】**
 - A) 4次
 - B) 5次
 - C) 6次
 - D) 无限次
- 文件包含预处理中被包含的文件应该是 **【 】**
 - A) 目标文件
 - B) 源文件
 - C) 可执行文件
 - D) 批处理



习题——选择题

- 设有说明： `int a,b,*x,*y;` 及函数 `sub(int *p, int *q)`
`{int t; t=*p; *p=*q; *q= t;}`, 则能够交换变量 `a,b`
的值的函数调用语句为 【 】
 - A) `sub(a,b);`
 - B) `*p=a; *q=b; sub(p,q);`
 - C) `sub(p,q);`
 - D) `sub(&a,&b);`
- 有一个二维数组 `s[3][4]`, 其第2行第3列元素的正确表示是其第3行第4列元素的正确表示是
(`s[0][0]`是第1行第1列元素) 【 】
 - A) `*(*(s+2) +3)`
 - B) `*(s+2) +3`
 - C) `s[2,3]`
 - D) `s[3][2]`




习题——选择题

- main函数的正确说明形式是【 】
 - A) `main(int a, char *s)` B) `main(int a, char s[][])`
 - C) `main(int a, char s)` D) `main(int a, char *s[])`
- 要定义只可在所在源文件中使用的全局变量，则该变量的类别是【 】
 - A) `auto` B) `extern`
 - C) `static` D) `register`




习题——选择题

- 有一个二维数组s[3][4], 则*(*s+2)+3)代表的元素是 【 】
A) s[2,3] B) s[2][3] C) s[3,4] D) s[3][4]
- 设有说明: int a,b,*p,*q; 及函数sub(int *p, int *q) {int t; t=*p; *p=*q; *q= t;}, 则能够交换变量a,b的值的函数调用语句为 【 】
A) sub(a,b); B) *p=a; *q=b; sub(p,q);
C) p=&a;q=&b;sub(p,q); D) sub(p,q);




习题——选择题

- 若用数组名作为函数调用的实参，传递给形参的是【 】
 - A) 数组第一个元素的值
 - B) 数组中全部元素的值
 - C) 数组的首地址
 - D) 数组元素的个数
- 不能作为sizeof()的运算对象的是【 】
 - A) 变量名
 - B) 简单类型名
 - C) 结构类型名
 - D) 算子名称



习题——选择题

- 若有定义 `int a[5], *p=a`，则对 `a` 数组元素的正确引用是：【 】
 - A) `*&a[5]`
 - B) `a+2`
 - C) `*(p+5)`
 - D) `*(a+2)`
- 编译程序对宏命令的处理是【 】
 - A) 在程序运行时进行
 - B) 在程序连接时进行
 - C) 编译之前
 - D) 编译之后;



习题——选择题

- 函数`rewind()` 的作用是【 】
 - A) 使文件位置指针重新返回文件开头
 - B) 使文件位置指针指向文件中所要求的特定位置
 - C) 使文件位置指针指向文件末尾
 - D) 使文件位置指针自动移至下一个字符位置;
- 设`int a[9], *p=a;` 则不能表示`a[1]`地址的是【 】
 - A) `p+1`
 - B) `a+1`
 - C) `a++`
 - D) `++p`



习题——选择题

- 设 $a=1, b=2$, 则表达式 $a \ll b$ 的值是【 】
A) 1 B) 2 C) 3 D) 4
- C语言中运算对象必须是整型的运算符是【 】
A) $\% =$ B) $/$ C) $=$ D) $*$
- 设 $\text{int } a=1, b=2, x$; 则表达式 $x = a + (a > b ? a : b)$ 的值是【 】
A) 0 B) 1 C) 2 D) 3



习题——选择题

- 设 `int x=1,y=3`; 执行后 `x` 的值不等于 6 的赋值语句是 【 】
 - A) `x=(x=1+2,x*2);`
 - B) `x=y>2?6:5;`
 - C) `x=9-(--y) -(--y);`
 - D) `x=y*4.6/2;`
- 结构化程序设计所规定的三种基本控制结构是 【 】
 - A) 输入,处理,输出
 - B) 树形,网形,环形
 - C) 顺序,选择,循环
 - D) 主程序,子程序,函数




习题——选择题

- 与`if(*p++==*q++)a=b;`不等价的是 【 】
 - A) `if(*(p++)==*(q++))a=b;`
 - B) `if((*p++)==(*q++))a=b;`
 - C) `if((*p)++==(*q)++)a=b;`
 - D) `if(*p==*q) {a=b;p++;q++;}`
- 设`char a[]="abcdef;"` `b[]={‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’};`则不正确的叙述是 【 】
 - A) a与b完全相同
 - B) a与b规模相同
 - C) a与b都存放字符串
 - D) a比b规模大



习题——选择题

- 判断字符串s是否大于字符串t应使用 【 】
A) if (s>t) B) if (strcmp (s, t) >0)
C) if (strcmp (t, s) >0) D) if (strcmp (s,t))
- 定义int (*p)[4]中p的含义是 【 】
A) 指向整型变量的指针
B) 指针数组名
C) 指向含4个分量的数组的指针
D) 定义不合法




习题——选择题

- 设有定义 `int a[10]`，则 `p+5` 表示 【 】
 - A) `a[5]` 的地址
 - B) `a[6]` 的地址
 - C) `a[5]` 的值
 - D) `a[6]` 的值



习题——选择题

- 若有定义：`struct student { int age ;
int num ; }std , *p; p=&std ;`则对以下结构体变量std中成员age的引用方式不正确的是【 】
 - A) `std. age`
 - B) `p->age`
 - C) `(*p) .age`
 - D) `student-> age`
- `sizeof(float)`是【 】
 - A) 函数调用
 - B) 整型表达式
 - C) 浮点表达式
 - D) 无意义



习题——选择题

- 设`int b=1, a=b>>2;` 则【 】
 - A) `a=b`
 - B) `a=2b`
 - C) `a=4b`
 - D) `a=b/2`
- 用`fopen`打开一个新的可以读写的二进制文件，则文件的方式字符串为【 】
 - A) `"ab+"`
 - B) `"wb+"`
 - C) `"rb+"`
 - D) `"ab"`




习题——选择题

- fscanf的正确调用格式是【 】
 - A) fscanf（文件指针，格式串，输出列表）；
 - B) fscanf（文件指针，格式串，输入列表）；
 - C) fscanf（格式串，文件指针，输入列表）；
 - D) fscanf（格式串，文件指针，输出列表）；
- 函数的返回类型由【 】
 - A) return中的表达式类型决定
 - B) 计算机随机定义
 - C) 函数首部中函数类型决定
 - D) 由主调函数决定



习题——选择题

- 设 `int b, a; float c; c=(float)a+b;` 则 `(float)` 的作用是使 【 】
 - A) 变量 `a` 的类型发生改变
 - B) 变量 `a, b` 的类型发生改变
 - C) 变量 `a` 的值的类型发生改变
 - D) 变量 `a, b` 的值的类型发生改变
- C语言中，二维数组元素在内存中的存放顺序是 【 】
 - A) 按行存放
 - B) 按列存放
 - C) 由用户自己定义
 - D) 无规律



习题——选择题

- 设有说明：char
*lag[]={"Fortran","Basic","Pascal","Java",
"C"}; 则表达式:*lag[1] > *lag[3]比较的是
【 】
 - A) 字符F和字符P
 - B) 字符串Basic和字符串Java
 - C) 字符B和字符J
 - D) 字符串Fortran和字符串Pascal



习题——阅读题

```
#include<stdio.h>
main()
{ char a[ ]="-123";
  int k,r=5,flag,m=0;
  if (a[0]=='+'||a[0]=='-')
    flag=(a[0]=='+') ? 1:-1 ;
  for(k=1;a[k]!='\0';k++)
    m=m*r+a[k]-'0' ;
  printf("result=%d",flag*m);
} 【          】
```



习题——阅读题

```
#include<stdio.h>
main()
{ char a[ ]="language", b[ ]="program";
  char *p=a ,*q=b;
  while(*p!='\0')
  {
    if(*p++==*q++) break;
    printf("%c",*p);
  }
} 【          】
```



习题——阅读题

```
#include<stdio.h>
main()
{int b[4][4]={1,2,3,4},{5,6,7,8},
{9,10,11,12},{13,14,15,16}};
  int *p,(*w)[4];
    p=b[0]; w=b;
    printf("%d,",*p+2); p++;
    printf("%d,",*(*w+2)+2)); w++;
    printf("%d,",++(*p));
    printf("%d,",*(*w+3));
    printf("%d,",*(b[3]+3));
} 【          】
```



习题——阅读题

```
#include <string.h>
main()
{char *p1; char *p2;
char str[30]= "xyz";
  p1="abcd";
  p2="ABCD";
  printf("%s\n",strcat(p1,p2));
  strcpy(p1,str);
  printf("%s",str);
} 【      】
```




习题——阅读题

```
#include<stdio.h>
main ()
{ int x=1, y=0, a=0, b=0;
  switch (x)
  { case 1: switch (y)
            { case 0: a++; break;
              case 1: b++; break;
            }
    case 2: a++; b++; break;
    case 3: a++; b++;
  }
  printf ("a=%d, b=%d\n", a, b) ;
} 【          】
```



习题——阅读题

```
#include <stdio.h>
main( )
{ sub( int );
  int  n = 5;
  printf("%d\n",sub(n));
}
sub( int  n )
{ int  a;
  if( n == 1) return  1;
  a = n + sub( n-1 );
  return( a );
} 【          】
```



习题——阅读题

```
#include<stdio.h>
main()
{
int a[3][3]={{3,2,2},{4,5,6},{-2,9,2}};
int i, j, s=0;
for(i=0; i<3; i++)
for(j=0; j<3; j++)
if(i == 2-j) s=s+a[i][j];
printf("s=%d",s);
} 【          】
```




习题——阅读题

```
#include<stdio.h>
main( )
{ int x=20, y=15;
  if( x>y) swap(&x, &y);
  printf("%d", x);
}
swap(int *p1, int *p2)
{ int *temp;
  temp=p1;
  p1=p2;
  p2=temp;
} 【      】
```



习题——阅读题

```
#include<stdio.h>
main()
{ int i=0;
  do
  { i++;
    if(i%2==0) continue;
    if (!(i%3)) printf("%d,",i);
  }while(i<=10);
} 【          】
```


$$s = 3.14 + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \frac{1}{5!} + \dots$$

习题——编程题

求;


$$s = 3.14 + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \frac{1}{5!} + \dots$$

直到最后一项的绝对值小于某个给定的充分小的正数。

求;

$$s = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots - \frac{1}{4n-3} + \frac{1}{4n-1}$$


直到最后一项的绝对值小于某个给定的充分小的正数。



习题——编程题


- 定义函数求：

$$s(n, x) = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \cdots + (-1)^{n+1} \frac{x^n}{n!}$$



习题——编程题

- 任意读入100个数，编程将它们排序打印并给出排序前的位置。
- 求1~n（任意给定）之间所有包含5的素数。



习题——编程题

- 从键盘读入一批字符，分别统计其中数字的出现次数、大、小写字母的出现次数，并分别按次数大小排序输出对应的字符及次数。

习题——编程题

编程打印图形

A

C b A


A b C d E

G f E d C b A

A b C d E f G h I

o o o o o o o o

Z y X , , , o o o o C b A



习题——编程题

编程打印图形

```
A
 B C D
  E F G H I
   J K L M N O P
    Q R S T U V W X Y
     J K L M N O P
      E F G H I
       B C D
        A
```