

# 上海市高等学校信息技术水平考试

## C 程序设计及应用（二三级）

### 考试大纲（2025 年版）

#### 一、考试性质

上海市高等学校信息技术水平考试是上海市全市高校统一的教学考试,是检测和评价高校信息技术基础教学水平和教学质量的重要依据之一。该项考试旨在规范和加强上海高校的信息技术基础教学工作,提高学生的信息技术应用能力。考试对象主要是上海市高等学校在校学生。考试每年举行一次,通常安排在当年十一月底的星期六或星期日。凡考试成绩达到合格者或优秀者,由上海市教育委员会颁发相应的证书。

本考试由上海市教育委员会统一领导,聘请有关专家组成考试委员会,委托上海市教育考试院组织实施。

#### 二、考试目标

程序设计及应用是基于不同语种的程序设计水平考试。通过程序设计的教学和考核,旨在提高大学生的计算思维和编程能力,重在培养大学生应用程序设计语言编写程序解决工程实际问题的能力。考试现有 C、Java、Python 三个语种,根据掌握的知识和能力分二、三两个等级。

C 程序设计及应用二级的目标是考核学生掌握并能应用基本知识解决数据类型、基本语句、模块化程序设计、常用算法、函数、文件等简单的实际问题;三级的目标是在二级的基础上,增加数据处理、科学计算、趣味解题、综合应用等应用场景的理解与抽象,面向学科交叉解决较复杂实际问题的能力。

#### 三、考试内容和要求

知识领域	知识单元	知识点	要求
C 程序结构与运行	程序结构	程序代码结构：预处理、全局声明、函数定义及调用	掌握
		语法组成规则：单词、表达式、语句、函数、程序	掌握
		词法构成规则：标识符、常量、运算符	掌握
		多源程序组成及外部数据	知道
	编程环境	编程环境的选择及操作使用	掌握
		程序的编辑、编译、运行	掌握
		程序文件命名及相关扩展名：.c、.exe 等	掌握
		程序正确性调试：测试用例、变量观察、跟踪、断点	掌握
		程序编译参数与编译命令	知道
	预处理命令	文件包含命令：#include	掌握
		宏定义及符号常量：#define	掌握
		带参数的宏定义：#define ()	理解
		条件预处理：#ifdef、#else、#endif 等	知道
C 语言基础要素	数据类型	基本类型：整型、浮点型、字符型、字符串	掌握
		类型扩展：单双精度、长短整型、无符号带符号	掌握
		各类型数值范围、长度及存储格式	掌握
		数据类型综述及数据类型选择	掌握
	数据常量	整型常量、浮点型常量、字符型常量、字符串常量	掌握
		数字字符、大小写字母的编码关系（ASCII 编码）	掌握
		常量表示的前缀与后缀	理解
		常量对应的数据类型	掌握
	变量	变量声明及初始化	掌握
		变量作用域：全局变量、局部变量、复合语句	掌握
		变量生命期：变量创建与销毁	理解
	基本运算	算术运算：+ - * / % ++ --	掌握
		关系运算：< <= > >= == !=	掌握

知识领域	知识单元	知识点	要求
		逻辑运算及运算优化：&&    !	掌握
		位运算：&   - ^ << >>	理解
		赋值运算(=)与复合赋值(+=等)	掌握
		条件运算(?:)	掌握
		逗号运算(,)	理解
		强制类型转换运算:(int) 等	掌握
		存储空间计算运算: sizeof()	掌握
		运算符的优先级及结合性	掌握
	表达式	表达式组成规则及求值顺序	掌握
		表达式的逻辑真假判定	掌握
		自动类型转换	掌握
		表达式书写与计算	掌握
	语句	语句综述: 声明语句、执行语句、空语句、复合语句	掌握
		输入输出语句及格式符: printf, scanf	掌握
		顺序结构: 数据交换示例	掌握
		选择结构语句: if, switch	掌握
		选择语句嵌套	掌握
		循环结构语句: for, while, do-while	掌握
		循环语句嵌套	掌握
		控制转移语句: return, break, continue	掌握
C 语言核心要素	数组	批量数据的组织	理解
		一维数组的声明、初始化及引用	掌握
		一维数组在函数间传递	掌握
		一维数组与字符串	掌握
		二维数组的声明、初始化及引用	掌握
		二维数组在函数间传递	理解

知识领域	知识单元	知识点	要求
C 语言高级要素	函数	函数声明：函数类型、函数名、形式参数表	掌握
		函数定义：函数声明、函数体	掌握
		函数返回：返回语句、返回表达式、空类型 void	掌握
		函数调用：实际参数、参数传递、传值与值址	掌握
		函数的嵌套调用及运行流程	掌握
		全局变量与局部变量	掌握
		变量存储类别、变量声明前缀、存储区分类	理解
		递归函数及递归设计	理解
		用户函数库、用户.h	理解
		主函数的命令行参数	知道
	指针	变量概念：类型、名称、值、（首）地址、长度	掌握
		指针变量的声明及初始化	掌握
		指针运算：取址运算&、取值运算*	掌握
		指针法与下标法的等价性	掌握
		使用指针优化代码效率	理解
		字符串处理（字符数组与字符指针）	掌握
		字符串函数：gets, puts, strlen, strcpy, strcat, strcmp	掌握
		使用指针管理存储空间：malloc, free	掌握
		指针数组、多字符串处理	理解
		二级指针	理解
		指向一维数组的指针	知道
		函数指针、函数指针数组及调用	知道
	结构体与共用体	结构体类型的声明：struct	掌握
		结构体变量的声明、初始化及引用	掌握
		结构体数组的声明、初始化及引用	掌握
		结构体指针的声明、初始化及引用	掌握

知识领域	知识单元	知识点	要求
		结构体在函数间传递与返回	理解
		单向链表的建立与管理，结点的插入与删除	理解
		共用体类型及使用：union	理解
		枚举类型及使用：enum	理解
		类型重定义：typedef	理解
		复杂数据结构的复合声明	知道
	文件	文件类型：文本文件、二进制文件	掌握
		文件指针变量：FILE *	掌握
		文件的打开和关闭：fopen, fclose	掌握
		文件读写：fprintf, fscanf, fgetc, fputc, fgets, fputs	掌握
		二进制文件操作：fread, fwrite	理解
		文件定位函数：feof, fseek, ftell, rewind	理解
		其他文件函数：ferror 等	知道
程序思维 及算法	常用表达式	常用表达式的书写与计算	掌握
		计算 M 进制数的第 N 位，或小数点后第 N 位	掌握
		小数点后某位截去尾数或四舍五入	掌握
		判断数值处于某些区间内或区间外	掌握
		判断多数连续递增、连续递减、或最大最小值	掌握
		判断是否除零、是否整除、是否奇偶、是否闰年	掌握
		判断是否数字字符、大写字母、小写字母、字符串结束符	掌握
		数字字符与数值转换、大小写字母转换	掌握
	常用库函数	常用系统库函数及对应头文件	掌握
		字符输入输出：getchar, putchar	掌握
		数学函数：sqrt, sin, cos, fabs, pow, exp, log	掌握
		随机数函数：srand, rand	理解
		系统函数：exit, system	理解

知识领域	知识单元	知识点	要求
	常用算法	常用算法的算法理解与代码编写	掌握
		累加累乘、最大最小、次大次小	掌握
		排序（冒泡法/选择法/插入法）	掌握
		查找（顺序查找/二分法）	掌握
		复合数据的排序：多字符串、多字段	理解
		批量数据的遍历、筛选、统计、转换、移位等	掌握
		是非问题：素数判断、对称判断等	掌握
		计算问题：进制转换、级数累加等	掌握
	代码风格与书写规范	程序代码风格良好、书写规范	掌握
		排版整洁：缩进、空格舒展、每行一条语句	掌握
		易于阅读：注释、功能分段、变量不混用、命名有意义	掌握
		界面友好：有提示有帮助不崩溃、输入检查、输出对齐	掌握
		代码重用：代码段或函数可以重复使用且容易修改升级	理解
	算法设计	按程序或模块的功能要求进行算法设计	理解
		算法概念、伪代码或流程图算法表示	理解
		结构化算法设计、数据组织合理有效	理解
		复杂模块化程序设计：自底向上、自顶向下	知道
	设计拓展	阅读拓展材料进行算法及程序设计	理解
		阅读应用背景说明，正确转化为程序问题	理解
		查阅库函数手册或使用范例，正确调用库函数	理解
		阅读外部资源及程序范例，正确理解并修改程序代码	理解
		面向综合应用的程序设计	知道

备注：

1. 对知识和技能的考核要求中，二级为知道/理解/掌握，三级全部为掌握。
2. 知识与技能的学习考核要求分为**知道**、**理解**和**掌握**三个层次，其含义分别为：  
**知道**：能识别和记忆相关的学习内容，对相关知识有初步认识。

**理解**:初步把握学习内容的表达、作用和使用方法,并能基于相应的学习内容完成简单编程。

**掌握**:以某一学习内容为核心,综合运用其他相关内容,实现给定问题的编程。

#### 四、试卷结构

题号	题型	题量(3个专题任选一个专题)		分值	考核内容	考核能力
		专题1各小题相对独立	专题2与专题3的主题,包括数据处理类、趣味解題类、综合应用类等			
第1问	单选题	基础题,10小题		15	基本概念 语法语义 专题理解	基本语言规范 持续学习能力 计算思维能力
第2问	程序题	程序填空题,1至2小题	围绕专题的程序题,每问1题,形式包括:程序调试、程序填空、程序编写等	20	程序理解 程序调试 规范优化	程序设计能力 程序调试能力 团队协作能力
第3问	程序题	程序调试题,1至3小题		30		
第4问	编程题	一般编程题,1至2小题	专题相关的二级编程题,1小题	35	常用算法 应用设计	程序设计能力 算法设计能力
第5问	编程题	三级编程题,1小题	专题相关的三级编程题,1小题	20	算法提升 综合应用	算法设计能力 挑战创新能力
第6问	编程题	三级综合题,1小题	专题相关的三级综合题,1小题	30		
合计		15至19题	15题	150分		

#### 五、相关说明

1. 考试时间:150分钟。
2. 试卷总分:150分。
3. 等第:不合格、二级合格、二级优秀、三级合格、三级优秀。各等第分数线由考委会划定。
4. 考试方式:考试采用基于网络环境的无纸化上机考试。
5. 考试环境:

- 上海市高等学校信息技术水平考试通用平台。
  - 操作系统：Windows 10 中文版。
  - 程序开发环境：Visual Studio 2012 及以上中文版( Visual C++组件 ), CodeBlocks 17.0 及以上, DevC++ 5.11 及以上, GUIDE 1.0 及以上。
6. 建议学时数： 48-64 学时，其中实验课不少于 20 学时。
7. 参考教材：
- 顾春华主编. 程序设计方法与技术—C 语言（第 2 版）. 高等教育出版社. 2024 年
  - 谭浩强主编. C 程序设计（第五版）. 清华大学出版社. 2017 年

## 六、题型示例

### 单选题

【例 1】中国居民身份证由 18 个字符组成，其中前 17 位为数字字符，第 18 位为数字字符或者罗马数字字符 X。为了在程序中方便处理居民身份证的输入输出和校验等，最合适的身份证变量声明应该是\_\_\_\_\_。

- A. char id[17];      B. char id[18];      C. char id[19];      D. long id;

【参考答案】C

【知识内容】数据类型：数据类型选择；数组：一维数组与字符串。

【能力目标】掌握各数据类型及类型选择，考核基本语言规范；理解 C 语言中字符串处理方法，考核程序设计能力。

### 程序填空题

【例 2】中国居民身份证的第 7 至第 14 位为居民出生日期，格式为“yyyymmdd”。程序输入一个身份证号码，提取其中的出生日期，并判断该日期是否有效。结构体 birth 为日期结构体，包含年、月、日信息。函数 getBirth 从身份证号码 id 中提取出生日期至 p 指针。函数 isIn 判断整数 n 是否大于等于 min 且小于等于 max，返回 1（是）或 0（否）。程序代码及运行结果如下，请填写 4 个空白使程序完整。

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

```
struct birth
```



```

{ //日期结构体
    int year;    //年
    int month;   //月
    int day;     //日
};

void getBirth(struct birth *p, char id[])
{ //从身份证号码 id 中提取生日至 p, id[6]至 id[13]对应年月日
    p->year = (id[6]-'0')*1000 + (id[7]-'0')*100 +
              (id[8]-'0')*10 + (id[9]-'0')*1;
    p->month = (id[10]-'0')*10 + (id[11]-'0');
    p->day    = (id[12]-'0')*10 + (id[13]-'0');
}

int isIn(int n, int min, int max)
{ //判断整数 n 是否在区间[min,max]中
    if (_____( 1 )_____)           //填空 1, 表达式
        return 1;
    else
        return _____( 2 )_____; //填空 2, 返回值
}

int main(void)
{
    char id[80];
    _____( 3 )_____;           //填空 3, 声明变量
    gets(id);
    getBirth(_____( 4 )____);       //填空 4, 调用函数, 传递参数
    printf("%d 年%d 月%d 日 ", b.year, b.month, b.day);
    if (isIn(b.month,1,12) && isIn(b.day,1,31))
        printf("生日有效\n");
}

```

```

else
    printf("生日无效\n");
return 0;
}
//输入：11010519491231002X，输出：1949 年 12 月 31 日 生日有效
//输入：123456789012345678，输出：7890 年 12 月 34 日 生日无效

```

#### 【参考答案】

- (1) 【`n>=min && n<=max`】           //填空 1，表达式  
 (2) 【0】                               //填空 2，返回值  
 (3) 【`struct birth b`】               //填空 3，声明变量  
 (4) 【`&b,id`】                       //填空 4，调用函数，传递参数

#### 【知识内容】

- (1) 常用表达式：判断数值处于区间内；  
 (2) 函数：函数返回表达式；  
 (3) 结构体：结构体变量的声明；  
 (4) 结构体：结构体在函数间传递与返回。

#### 【能力目标】

- (1) 掌握表达式的书写，考核基本语言规范能力；  
 (2) 掌握函数的返回及返回表达式，考核程序设计能力；  
 (3) 掌握结构体变量的声明，考核程序设计能力；  
 (4) 理解结构体与字符串在函数间的传递与返回，考核程序设计能力。

### 调试改错题

【例 3】中国居民身份证的第 18 位称为“检验位”，如果已经知道一个身份证号码的前 17 位，可以计算出身份证的第 18 位，也可以验证第 18 位是否正确，检验位的计算方法如图。

以下程序输入一个身份证号码，根据前 17 个字符检验第 18 位是否正确。程序代码包含了 3 个错误，找出错误点并加以改正。

【例】某男性的身份证号码为【53010219200508011X】，我们看看这个身份证是不是合法的身份证。

位置 i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
身份证各位 $a_i$	5	3	0	1	0	2	1	9	2	0	0	5	0	8	0	1	1
系数 $w_i$	7	9	10	5	8	4	2	1	6	3	7	9	10	5	8	4	2

(1)求和:

$$\sum(w_i * a_i) = (5*7) + (3*9) + (0*10) + (1*5) + (0*8) + (2*4) + (1*2) + (9*1) + (2*6) + (0*3) + (0*7) + (5*9) + (0*10) + (8*5) + (0*8) + (1*4) + (1*2) = 189$$

(2)求余:  $\sum(w_i * a_i) \% 11 = 189 \% 11 = 2$

(3) 根据余数找对应的校验位:

余数	0	1	2	3	4	5	6	7	8	9	10
对应的校验位	1	0	X	9	8	7	6	5	4	3	2

余数 2 对应的检验码是 X，因此该身份证号码是正确的。

改错方法：不允许增加或删除语句，但可以修改语句，所修改的语句必须在该句尾加上注释标记：“/\*\_\*/”或“//\_”。

```
#include <stdio.h>

int main(void)
{
    char id[80];

    int w[]={7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2}; //系数
    char last[]="10X98765432"; //余数 0 - 10 对应的校验码
    int i, k, sum=1; //①初始化 1 改为 0
    gets(id);
    for (i=0;i<18;i++) //②边界 18 改为 17
        sum += id[i] * w[i]; //③id[i]改为(id[i]-'0')
    k = sum % 11;
    if (id[17]==last[k])
        printf("最后一位正确\n");
    else
        printf("最后一位错误\n");
    return 0;
```

}

//输入：11010519491231002X，输出：最后一位正确

//输入：123456789012345678，输出：最后一位错误

**【参考答案】**

(1)【0】

(2)【17】

(3)【(id[i]- '0')】

**【知识内容】**

(1) 变量：变量声明及初始化；

(2) 语句：循环结构语句；

(3) 常用表达式：数字字符与数值转换；

(4) 设计拓展：阅读外部资源及程序范例，正确理解并修改程序代码。

**【能力目标】**

(1) 掌握变量声明及初始化，考核基本语言规范、程序调试能力；

(2) 掌握 for 循环的边界条件，考核基本语言规范、程序调试能力；

(3) 掌握数字字符转换为数值，考核基本语言规范、程序调试能力；

(4) 理解新计算方法，转化为正确的程序，考核持续学习能力。

## 编程题

**【例4】**编写程序，以字符串方式输入一个居民身份证号码，通过以下步骤初步检查该号码是否有错误，输出“初检通过”或“初检错误”。

(1) 字符串是否 18 位长度。

(2) 字符串前 17 位字符是数字字符。

(3) 字符串第 18 位字符是数字字符，或者是字母“X”。

**【知识内容】**

(1) 编程环境：程序的编辑、编译、运行；

(2) 数组：一维数组与字符串；

(3) 常用表达式：判断是否数字字符。

【能力目标】

- (1) 掌握编程开发环境的使用，能编写简单的程序并运行，考核基本语言规范；
- (2) 掌握字符串的输入和处理，掌握各字符类型的判断，考核程序设计能力。

【例 5】综合以上例 1 至例 4，编写程序，从文本文件“person.txt”中读取不超过 100 个居民身份证号码及姓名，尽可能检验身份证号码的有效性，如果无效设置出生日期为“0 年 0 月 0 日”，输入文本和输出结果如图，遵照“程序代码编写规范”要求，按以下顺序逐步完成程序。



- (1) 将例 4 程序的判断功能转化为函数，对身份证号码做初步检查。`int idChkBase(char id[];`
- (2) 将例 3 程序的判断功能转化为函数，检查身份证号码的最后一位。`int idChkLast(char id[];`

- (3) 将例 2 程序的出生日期检查转化为函数，函数声明：

`int chkBirth(struct birth *p); //返回 1 通过检验`

函数判断有效出生年份应该在 1900 至 2099 之间，并能判断出生当年是否闰年，计算出生当月的天数，从而判断出生当日的日期是否有效。

- (4) 设计结构体 `person` 如下，包含身份证号码、姓名和出生日期。

```
struct person
{
    char id[20];
    char name[20];
    struct birth b;
};
```

(1) 设计函数, 从文本文件 “person.txt” 中读取身份证信息, 保存在 ps 数组中, 返回成功读取的号码数。

```
int read(struct person ps[]); //返回读取成功的身份证数
```

(2) 设计函数, 调用 getBirth 函数提取 ps 数组中各身份证号码的出生日期, 调用 idChkBase、idChkLast、chkBirth 等函数检查身份证号码是否有效, 如果无效, 出生日期年月日全部置 0。返回有效身份证号码的个数。

```
int chkAll(struct person ps[], int n); //n 为元素个数
```

(3) 设计函数, 输出所有身份证信息, 输出格式如图所示。

```
int prtAll(struct person ps[], int n); //n 为元素个数
```

(4) 设计主函数, 调用上述各函数, 实现编程要求。

身份证	姓名	出生年月
11010519491231002x	zhangli	1949年12月31日
440524198001010013	wangxiao	1980年1月1日
310115200101010114	zhaolin	0年0月0日
53010219200508011x	qianli	1920年5月8日
110105198912150036	lilin	1989年12月15日
110203199310100047	lilin	1993年10月10日
310115201001010010	sunli	2010年1月1日

#### 【知识内容】

- (1) 代码风格与书写规范: 程序易于阅读, 函数代码可重用;
- (2) 算法设计: 模块化程序设计思想, 自底向上复杂程序设计;
- (3) 表达式: 判断闰年, 书写表达式; 数组: 一维数组初始化;
- (4) 结构体: 复杂数据结构的复合声明、结构体数组的初始化;
- (5) 文件: 文本文件读取; 结构体: 结构体在函数间传递与返回;
- (6) 函数: 函数的嵌套调用及运行流程;
- (7) 编程环境: 程序正确性调试 (测试用例)。

#### 【能力目标】

- (1) 理解易于阅读的代码风格和可重用的书写规范, 能将已成功的程序转化为可分享的函数, 考核团队协作能力;
- (2) 理解模块化设计思想, 理解自底向上的复杂程序设计, 考核算法设计能力;

- (3) 掌握各类表达式的书写，考核基本语言规范；
- (4) 理解复杂数据结构，考核程序设计能力。
- (5) 掌握文本文件的打开和读取、掌握结构体数组的初始化、理解结构体数组在函数间传递，考核程序设计能力；
- (6) 掌握函数之间的嵌套调用，考核程序设计能力；
- (7) 掌握复杂程序的正确性验证，考核程序调试能力。