C语言程序设计复习大纲

Review task

一、What's C?

1. Content

- [ ] C语言的历史沿革及特点：C语言的产生与发展及特点；
- [ ] C语言源程序的结构、书写规则与风格：源程序的组成、main函数和其他函数；头文件、数据说明、函数的开始和结束标志；输入与输出函数；C源程序的结构特点；源程序的书写格式、规则与风格；
- [ ] C语言的字符集、标识符与关键字：C语言的字符集与转义字符；C语言的标识符；C语言的关键字；
- [ ] C语言程序的开发环境：Turbo C 2.0集成开发环境；MS Visual C++ 6.0集成开发环境。

2. Requirements

- [ ] 了解C语言源程序的结构、书写规则与风格；C语言源程序的结构、书写规则与风格；C语言的转义字符、标识符定义与关键字；
- [ ] 理解C语言程序的开发过程，熟悉Turbo C 2.0集成开发环境与MS Visual C++ 6.0集成开发环境。

二、Algorithm.

1. Content

- [ ] 算法的概念：算法概念、特性；
- [ ] 如何表示一个算法：用自然语言表示算法；用流程图表示算法；三种基本机构和改进的流程图；用N-S流程图表示算法；用伪代码表示算法；用计算机语言表示算法；
- [ ] 结构化程序设计方法：自顶向下；逐步细化；模块化设计；结构化编码。

2. Requirements

- [ ] 了解算法的概念和特性；
- [ ] 了解描述算法的方式和方法；掌握用流程图描述算法的方法；掌握程序基本控制结构；会画简单的流程图；
- [ ] 了解结构化程序设计方法。

三、Data types, operators, and expressions.

1. Content

- [ ] C语言的数据类型分类：数据类型的基本概念；C语言数据类型分类；

- [ ] 常量与变量：常量与变量的定义、区别；常量与符号常量；符号常量的使用；
- [ ] 整型数据：整型常量的表示方法；整型变量的定义与分类；
- [ ] 实型数据：实型常量的表示方法；实型变量的定义与分类；
- [ ] 字符型数据：字符常量；转义字符；字符变量的蒂尼；字符数据在内存中的存储形式和使用方法；字符串常量；
- [ ] 变量的定义与赋初值；
- [ ] 各类数值型数据之间的混合运算：自动转换；赋值转换；强制转换；
- [ ] 运算符与表达式概述：运算符与表达式的分类；运算符的优先级；
- [ ] 算术运算符与算术表达式：基本算术运算符；负值运算符、自增运算符与自减运算符；算术表达式；
- [ ] 赋值运算符与赋值表达式：基本赋值运算符；复合赋值运算符；
- [ ] 逻辑运算符与逻辑表达式：逻辑运算符；逻辑表达式；
- [ ] 逗号运算符合逗号表达式：逗号运算符；逗号表达式；

2. Requirements

- [ ] 掌握C语言的基本数据类型；符号常量的定义；变量的定义与赋值；运算符的优先级；常用的运算符与表达式；不同类型数据的赋值转换与强制转换。

四、Sequential structure design.

1. Content

- [ ] C语句概述：C程序的结构；C语句；赋值语句；
- [ ] C语句简介：语句、语句标号及其使用；流程控制语句；表达式语句；函数调用语句；空语句；复合语句；
- [ ] 数据的输入与输出：输入、输出函数及其调用；格式输入与输出；字符数据输入与输出。

2. Requirements

- [ ] 掌握语句及其使用；复合语句；输入、输出函数及其调用（scanf，printf，getchar，putchar）；常用格式输入与输出（整型，浮点型，字符型，字符串);
- [ ] 理解函数调用语句、流程控制语句、表达式语句、空语句。

五、Branch structure design.

1. Content

- [ ] 关系运算符和表达式：关系运算符及其优先次序；关系表达式；
- [ ] 逻辑运算符和表达式：逻辑运算及其优先次序；逻辑表达式；
- [ ] if语句实现(if语句的3种形式与使用；if语句嵌套的二义性与解决)；
- [ ] 条件运算符和条件表达式；
- [ ] switch语句；

- [ ] 选择结构的嵌套。

2. Requirements

- [ ] 掌握用if语句、switch语句、条件运算符实现选择的方法；
- [ ] 掌握关系表达式和逻辑表达式的计算；
- [ ] 掌握选择结构的嵌套的方法。

六、Loop structure design.

1. Content

- [ ] 循环结构实现的几种方法：用if语句以及用goto语句构成循环；用for语句实现；用while语句实现；用do-while语句实现；
- [ ] 循环的嵌套；
- [ ] 循环的强制跳出与结束：用continue语句跳出本次循环；用break语句结束循环；用goto语句跳出或结束循环。

2. Requirements

- [ ] 掌握用for语句、while、do-while语句实现循环；
- [ ] 掌握循环的嵌套、强制跳出与结束；
- [ ] 掌握循环结束的条件、循环体执行次数的判断与计算，能写出循环执行的中间计算结果。

七、Array.

1. Content

- [ ] 数组概述：数组的概念与分类；
- [ ] 一维数组：定义形式、引用、初始化、输入与输出；
- [ ] 多维数组：二维数组的定义形式、引用、初始化；
- [ ] 字符数组与字符串：字符数组的定义形式、初始化、引用、输入与输出；字符串结束标记；常用的字符串处理函数。

2. Requirements

- [ ] 掌握一维数组、多维数组和字符数组的定义形式、初始化、存储与引用；存放字符串的字符数组的特殊性；
- [ ] 熟悉常用的字符串处理函数。

八、Function.

1.Content

- [ ] 函数概述：函数的概念与意义；模块化程序设计与函数；C语言函数的分类；
- [ ] 函数的定义形式：函数定义的一般形式；

- 函数的参数和函数的值：形式参数与实际参数；参数值的传递；函数的返回值；
- 函数的调用：函数调用的一般形式；函数调用的方式；被调用函数的声明与函数原型；文件包含与库函数的调用；
- 函数的嵌套调用；
- 数组作为函数参数：数组元素作为函数实参；数组名作为函数参数；值传递与地址传递；
- 局部变量和全局变量：局部变量、全局变量及其作用域；
- 变量的存储类别：动态存储方式与静态存储方式；static类型的变量。

2. Requirements

- 掌握C语言函数的意义、分类；函数定义的一般形式；有参函数的参数与参数值传递；函数调用的一般形式；数组元素与数组名作为函数参数的区别；文件包含与库函数的调用；
- 掌握局部变量、全局变量的概念、定义方法及作用域；理解变量的存储类别及其声明与生存期；变量声明与变量定义的区别与作用。。

九 、Precommand.

1.Content

- 概述：预处理的作用和使用；
- 宏定义：无参宏定义；带参宏定义；
- 文件包含：文件包含的含义与意义；文件包含命令的格式与使用注意；
- 条件编译：条件编译的含义与意义；条件编译命令的格式与使用注意。

2.Requirements

- 掌握无参宏定义、有参宏定义的方法与使用；文件包含命令的格式与使用；
- 理解编译命令的格式与使用注意。

十、Point.

1.Content

- 指针概述：变量地址、指针、指针变量的概念；地址运算符、指针运算符、指针变量标识符的作用与用途；指针变量的数据类型；指针变量的一般定义形式与赋值规则；指针变量的运算；
- 指针与简单变量：指向简单变量的指针及其指针变量的定义形式与赋值；指向简单变量的指针变量的引用；指针变量作为函数参数与简单变量作为函数参数的区别；
- 数组指针和指向数组的指针变量：指向一维数组的指针及其指针变量的定义形式与赋值；指向一维数组的指针变量的引用；指向一维数组的指针变量的运算；用指向数组的指针变量作为函数的参数；数组名作为函数参数的含义；
- 字符串的指针和指向字符串的指针变量：定义指向字符串的指针；使用字符串指针引用字符串；字符串指针变量与字符数组的区别；

2. Requirements

- [ ] 掌握变量地址、指针、指针变量的概念；地址运算符、指针运算符、指针变量标识符的作用与用途；指针变量的数据类型；指针变量的一般定义形式与赋值规则；
- [ ] 了解指向简单变量、一维数组和字符串的指针及其指针变量的定义形式、赋值、运算与引用；用指向变量、数组的指针变量作为函数的参数；用数组名作为函数的参数。

十一、Struct and union.

1.Content

- [ ] 构造数据类型概述：基本概念、共性与用途；结构体、共用体及枚举类型的特性与差别；
- [ ] 结构体类型：结构体的概念、定义方法、一般形式与成员初始化，赋值；结构体成员的引用方法；
- [ ] 枚举类型：枚举的概念、定义方法、一般形式与成员初始化，赋值；共用体成员的引用方法。

2.Requirements

- [ ] 掌握结构体的定义方法、一般形式、成员初始化及成员引用、赋值；枚举的定义方法、一般形式、成员初始化及成员引用、赋值。

Review notes

1. C语言在1971年在贝尔实验室开发出来，主要是用于UNIX系统的开发，具有较强的可移植性，如今发布了最新的ANSI C11版本。
2. 头文件的书写，#ifndef #define #endif；源文件书写，包含#include<xxx.h>, int main(), return 0;
3. 建议一行写一行代码，代码末尾用“；”表示结束。
4. 多写程序运行必要说明和解释==注释；/**/多行注释 和 //单行注释。
5. 标识符命名需要使用英文，适宜采用小写加下划线的命名方式。
6. “a”是字符串常量，’a’才是字符常量。
7. 宏常量 #define PI 3.15159。
8. 枚举常量 enum reponse{no = -1， yes = 1， none = 0， unsure = 2}；。
9. const的作用是定义一个不可被修改的常量。
10. char == 1bytes，int == 4bytes，short int == 2bytes，long int == 4bytes，float == 4bytes，double == 8。
11. a = 97， A = 65。
12. 在运算浮点数的时候要防止数据溢出，即代表数据有效位不能表示数据实际位。
13. 在C语言中，用非0就表示”真“，用0表示“假”。
14. 闰年计算公式“(year % 400 != 0 ||((year % 4 == 0) || (year % 100 != 0))”
15. 0000 1111 & 0000 0001 = 0000 0001;
16. 0000 1111 | 0111 1111 = 0111 1111;
17. 0000 0011 ^ 0000 0101 = 0000 0110;
18. ~0000 0101 = 1111 1010;
19. 两个或两个以上的语句序列叫做复合语句。
20. 自顶向下，逐步求精的方法的时候，如果当前方法未完成，可以采用空语句的方法。
21. getchar()的作用是从终端输入一个字符，按回车键表示输入结束。
22. putchar()的作用是向终端输出一个字符。

23. scanf(“%d”, &a);printf(“%d”, a); 。
24. printf中%%f输出为%f。
25. scanf(“%2d*%2d%2d”, &a, &b); input:123456; a = 12; b = 56;
26. 衡量算法的正确性：有穷性，确定性，有效性，0或n个输入，1或n个输出。
27. 程序的控制结构分为：顺序结构，循环结构，选择结构。
28. 程序流程图简单画法：椭圆形框称为开始/结束框，长方形框称为一般处理框，菱形框称为判断框，斜矩形框称为I/O框，连线称为流程线，圆形称为连接符。
29. 顺序结构主要由三个步骤组成：1.输入算法所需要的数据；2.进行数据处理；3.输出数据处理后的结果。
30. 选择结构主要是if else switch。
31. 白盒测试主要在测试的早期，尽量覆盖每一条分支语句。
32. 黑盒测试主要用在测试的后期，只求输入与输入，是否符合我们的预期。
33. 循环结构主要是 for, do-while,while。
34. 流程控制主要是goto(已弃用)，break，continue。
35. 函数内部接受的数据叫做形式参数，调用者提供的参数叫做实际参数。
36. 全局变量是指在函数任何地方定义都是有效的。
37. 局部变量指在函数内部定义才有效，函数外部如果不进行参数调用的话是无法修改这个变量值的。
38. static关键字指的是在语块内有效果，一旦定义，直到程序退出前都不会重复定义。
39. 文件中#include<stdio.h>和#include”stdio.h”，一个在标准库下面，一个在源文件下面查找。
40. 宏定义中#define和#undef是一对。
41. char arr[6] = “china”正确的，char arr[5] = “china”是错误的，因为数组的最后需要留一个\0作为数组终止符。
42. 数组的输出推荐使用for (i =0; str[i] != '\0’; i++) printf(“%c”, str[i]);
43. 定义一个变量类型确定其所占内存空间的大小，然后分配的内存空间的首地址，作为该变量的地址。而在变量所占存储单位中存放的数据，称为变量的值。
44. 指针是一个地址，而指针变量是存放地址的变量。
45. &是地址运算符，*是指针运算符。
46. (char *ptr = “china”) == (char ptr[6] = “china”)。
47. a = *(p++); 等价于 a = *p; p = p + 1;先取出p所指向的单元中的内容赋值给a，再使p只想下一个地址单元，这里指针变量p的指向发生了改变，而p所指向的存储单元中的内容并未发生变化。
48. a = (*p)++;等价于 a = *p; *p = *p + 1; 先取出p所指向的单元中的内容赋值给a，再使p所指向的单元中的内容加1，这里指针变量p的指向没发生变化，p所指向的存储单元发生了变化。
49. 用结构体实现链表。
50.