《C语言程序设计》复习提纲

**第1章 程序设计和C语言**

**C语言的注释(Annotation of C Language)**

* **注释是对程序的解释说明。**
* **Annotation is an explanations of the program.**
* **//：单行注释**
* **Single-Line Comment**
  + **/\*……\*/：多行注释**
* **multiline comments**

**C程序的上机步骤  
C program steps on the computer**

**1.输入和编辑源程序（.c文件）**

**Enter and edit source programs**

**2.对源程序进行编译**

**Compile the source program**

**3.运行可执行程序，(有时需要输入数据)得到运行结果**

**Run the executable program (sometimes you need to enter data) to get the results.**

**第2章 算法**

**算法的特性  
Characteristics of the algorithm**

**(1) 有穷性。一个算法应包含有限的操作步骤，而不能是无限的。**

**Fineness: An algorithm should contain limited operation steps, but not be infinite.**

**(2) 确定性。算法中的每一个步骤都应当是确定的，而不应当是含糊的。**

**Certainty: Every step in the algorithm should be deterministic (nirdharok), not ambiguous.(ospostho noy).**

**(3) 有零个或多个输入。**

**There are zero or more inputs.**

**(4) 有一个或多个输出。算法的目的是为了求解，“解” 就是输出。没有输出的算法是没有意义的。**

**There are one or more outputs. The purpose of the algorithm is to solve, and the "solution" is the output. Algorithms without output are meaningless.**

**(5) 有效性。算法中的每一个步骤都应当能有效地执行，并得到确定的结果。例如，如果b=0,则执行a/b是不能有效执行的。**

**Effectiveness. Every step in the algorithm should be able to execute effectively, and get a certain result. For example, if b = 0, a / b cannot be executed effectively(Because denominator cannot be 0).**

**三种基本结构Three basic structures**

**C语言是一种结构化程序设计语言。结构化程序有顺序结构、选择结构和循环结构三种基本结构。**

**C is a structured programming language. Structured program has three basic structures: sequence structure, selection structure and loop structure.**

第3章 简单数据处理

**变量(variable )**

**变量：变量代表内存中具有特定属性的一个存储单元，它用来存放数据，也就是变量的值，在程序运行期间，这些值是可以改变的。**

**Variables: Variables represent a storage unit with specific properties in memory. They are used to store data, that is, the values of variables, which can be changed during the running of the program.**

**Example:**

* + - **int a;**
    - **a=5; //Store 5 into a**
    - **a=a+5; //a的值变成10 (The value of a becomes 10)**

**变量必须先定义，后使用。**

* **Variables must be defined before they are used.**
* **定义变量时指定该变量的名字和类型。**
* **When defining a variable, specify the name and type of the variable.**

**Example:**

* + **int a; //a is an integer variable**
  + **char b; //b is a character variable**
  + **float c; //c is a single-precision decimal variable**
  + **double d; //d is a double precision decimal variable**

**标识符：一个对象的名字（如变量名、数组名、函数名等）**

**Identifier: The name of an object (such as variable name, array name, function name, etc.)**

* **Ｃ语言规定标识符只能由字母、数字和下划线3种字符组成，且第一个字符必须为字母或下划线。**
* **C language stipulates that identifiers can only be composed of letters, numbers and underscores, and the first character must be letters or underscores.**

**数据类型(data type)**

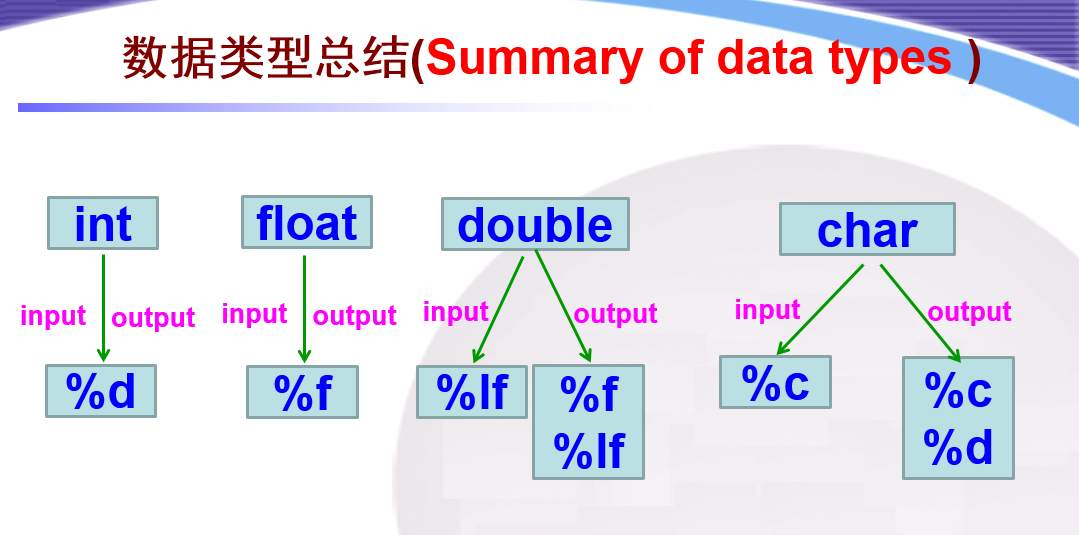
* **整型integer(int)**
* **example:10,100,123**
* **单精度浮点型single-precision float point(float)**

**\*example:3.14,5.0(Up to 6 decimal places)**

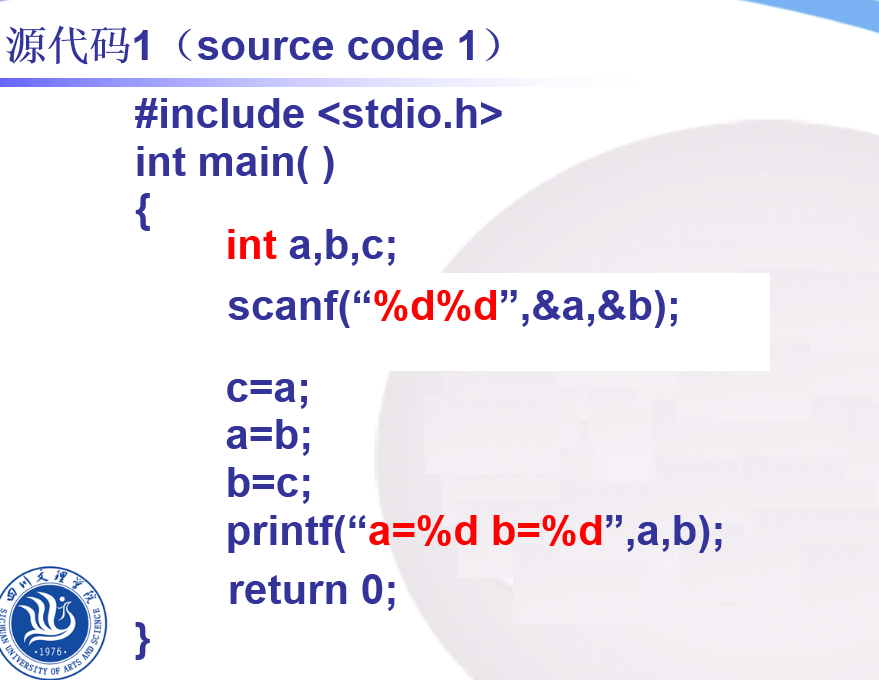
* **双精度浮点型double-precision float point(double)**
* **example:3.14159265358979**

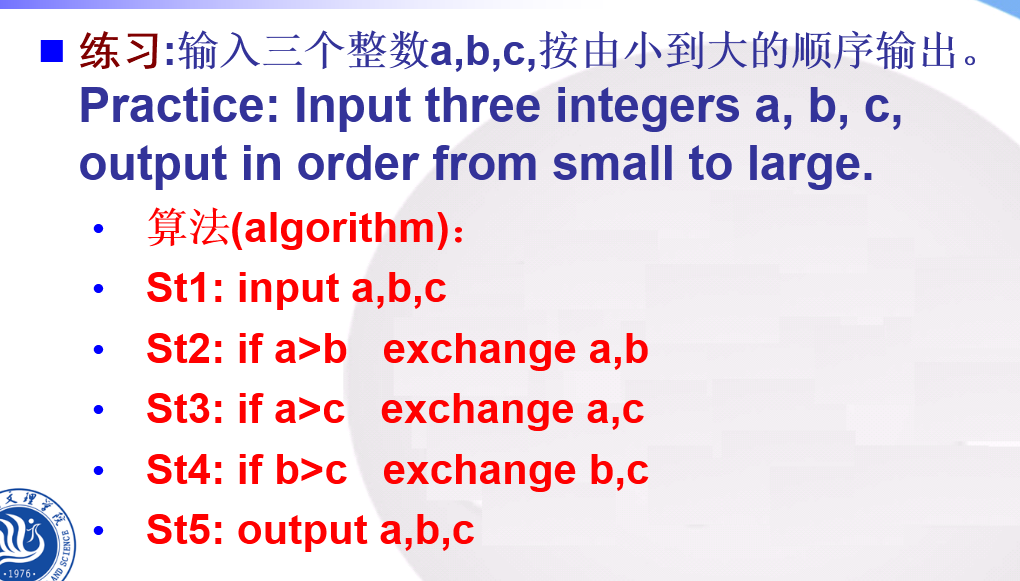
**(Up to 15 decimal places)**

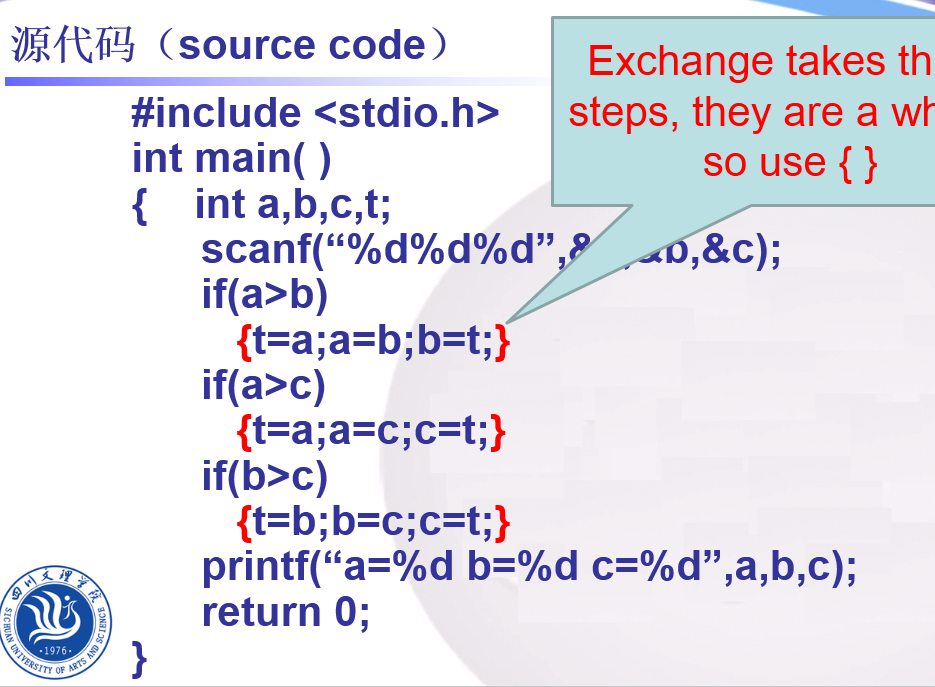
* **字符型character(char)**
* **每个字符使用单引号**
* **Use single quotes for each character**
* **example:**
  + **uppercase letter:'A' to 'Z'**
  + **Lower case letters:'a' to 'z'**
  + **Numeric characters:'0' to '9'**
  + **Other symbols:'@','?' etc.**
* **例3.3 给定一个大写字母，要求用小写字母输出，并输出小写字母的ASCII码值（十进制整数形式）。**
* **Example 3.4 Given a capital letter, it is required to output in lowercase and ASCII code value in decimal integer form.**
* **#include <stdio.h>**
* **int main ( )**
* **{**
* **char c1,c2;**
* **c1=’A’;**
* **c2=c1+32;**
* **printf("%c\n",c2);**
* **printf(“%d\n”,c2);**
* **return 0;**
* **}**



* **练习: 编写三个程序，分别将两个不同的整数、小数、字符交换**

**Exercise: write three programs to exchange two different integers, decimals, and characters.** 





**运算符和表达式（Operators and Expressions ）**

* **两个整数相除的结果为整数**
* **The result of dividing two integers is an integer.**
  + **如5/2的结果值为2（不是2.5），舍去小数部分**
  + **For example, the result value of 5/2 is 2（not 2.5）, and the decimal part is omitted.**
* **% 运算符要求运算对象为整数，结果也是整数。如8%3，结果为2**
* **% Operators require that the object of operation be an integer and the result be an integer. If 8% 3, the result is 2.**

**练习：编写程序输入一个整数，判断该数是否是偶数。  
Exercise: Write a program to enter an integer to determine if the number is even.**

**#include <stdio.h>**

**int main( )**

**{**

**int a;**

**scanf(“%d”,&a);**

**if(a%2==0)**

**printf(“YES\n”);**

**else**

**printf(“NO\n”);**

**return 0;**

**}**

* + **++i，--i：在使用i之前，先使i的值自加（减）1**
  + **Before using i, add (subtract) 1 of the value**
  + **i++，i--：在使用i之后，使i的值自加（减）1**
  + **After using i, add (subtract) 1 of the value**
* **示例：假设a=10;**
* **y=++a; y=11, a=11**
* **y=a++;y=10, a=11**
* **y=--a; y=9, a=9**
* **y=a--; y=10, a=9**

**在赋值符“＝”之前加上其他运算符，可以构成复合的赋值运算符**

**A compound assignment operator can be constructed by adding other operators before the assignment operator "="**

**ａ＋＝３等价于Equivalent to ａ＝ａ＋３**

**x\*=y+1等价于Equivalent to x＝x\*(y＋1)**

**printf函数**

**一般格式(General format)**

**printf（格式控制，输出表列）**

**例如：**

**printf(”i=%d,c=%c\n”,i,c);**

**example：**

* **int i=5;**
* **printf(”%d”,5); output:5**
* **printf(”%d”,i); output:5**
* **printf(”%d”,i+5); output:10**
* **i=i+5; printf(”%d”,i); output:10**

**格式字符（Format character）**

* + **%d 用来输出一个十进制整数(Used to output a decimal integer)**
    - **可以指定输出数据的列宽（You can specify the column width of the output data）**
  + **%ｃ 用来输出一个字符（Used to output a character）**

**char ch=’a’;**

**printf(”%c”,ch);**

**\*printf(”%-5c”,ch);**

**\*printf(”%5c”,ch);**

**scanf函数**

**scanf 函数的一般形式(General form)**

**scanf（格式控制，地址列表）**

scanf函数的格式字符（Format character）

与printf函数中的格式字符相似

Similar to format characters in printf functions

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

1. **选择结构程序设计**

关系运算符：

用来对两个数值进行大小比较的运算符(Used for comparison of two values)

Ｃ语言提供６种关系运算符(C language provides six relational operators)：

① ＜ (小于) ② ＜= (小于或等于)

③ ＞ (大于) ④ ＞= (大于或等于)

⑤ == (等于) ⑥ != (不等于)

关系表达式(Relational expression)

用关系运算符将两个数值或数值表达式连接起来的表达式

An expression that connects two numerical or numerical expressions with a relational operator

关系表达式的值是一个逻辑值，即“真”或“假”,在C语言中，以“1”代表“真”，以“０”代表“假”

The value of a relational expression is a logical value, namely "true" or "false". In C language, "1" represents "true" and "0" represents "false".

逻辑运算符(Logical Operator)

3种逻辑运算符(Three logical operators)：

&&（逻辑与）(Logic and)

||（逻辑或）(Logic or)

!（逻辑非）(Logical non)

逻辑表达式(Logical expression)

用逻辑运算符将关系表达式或其他逻辑量连接起来的表达式

Expressions that connect relational expressions or other logical quantities with logical operators

判断年龄age在13至17岁之内。

Age is between 13 and 17 years old.

age>=13 && age<=17

判断年龄age小于12或大于65。

Age less than 12 or more than 65.

age<12 || age>65

判断年龄age不是20岁（Age is not 20）。

age!=20

！（age==20）

逻辑表达式的值是“真”或“假”，在C语言中，以“1”代表“真”，以“０”代表“假”

The value of logical expression is "true" or "false". In C language, "1" stands for "true" and "0" stands for "false".

但在判断一个量是否为“真”时，以0代表“假”，以非0代表“真”

But in judging whether a quantity is true, 0 represents false and non-zero represents true.

条件运算符(Conditional operator)

一般形式(General form):

表达式１？表达式２: 表达式３

Expression 1?Expression 2:Expression 3

条件运算符的执行顺序：

The order of execution of conditional operators：

求解表达式1

Solving expression 1

若为非0（真）则求解表达式2，此时表达式2的值就作为整个条件表达式的值

\*If it is not 0 (true), then expression 2 is solved, and the value of expression 2 is taken as the value of the whole conditional expression.

若为0（假），则求解表达式3，表达式3的值就是整个条件表达式的值

If it is 0 (false), then expression 3 is solved, and the value of expression 3 is taken as the value of the whole conditional expression.

单分支选择if语句（Single Branch Selection）

例4.1 输入一个字符，如果是大写字母，则输出该字符，否则不做任何处理。

Example 4.1 Input a character, if it is a capital letter, then output the character, otherwise do not do any processing.

#include <stdio.h>

int main( )

{

char c;

scanf(“%c”,&c); // 输入一个字符

if(c>=‘A’&&c<=‘Z’) // 如果c是大写字母成立

printf (“%c\n”,c); // 输出c换行

return 0;

}

双分支选择if语句（Double Branch Selection）

例4.2 输入两个实数，按由小到大的顺序输出这两个数。

Example 4.2 inputs two real numbers and outputs them in the order of from small to large.

#include <stdio.h>

int main()

{ float a,b,t;

scanf("%f,%f",&a,&b);

if(a>b)

{ t=a;

a=b;

b=t;

}

printf("%5.2f,%5.2f\n",a,b);

return 0;

}

多分支选择if语句(Multi-branch selection)

例4.3 输入考试成绩，输出等级。（85分以上为优秀，60~84分为及格，其余为不及格）

Example 4.3 input test scores and output grades. (more than 85 points are good, 60~84 points are passed, and the rest are failed)

#include <stdio.h>

int main()

{ int score;

scanf("%d",&score);

if(score>=85)

printf("Good\n");

else if(score>=60)

printf("passed\n");

else

printf("failed\n");

return 0;

}

if语句的嵌套(Nesting of if statements)

在if语句中又包含一个或多个if语句称为if语句的嵌套

Including one or more if statements in an If statement is called nesting of if statements.

一般形式(General form )：

if( )

if( ) 语句1

else 语句2

else

if( ) 语句3

else 语句4

else总是与它上面最近的未配对的if匹配

Else always matches the most recent unpaired if on it

if ()

{if () 语句1 }

else语句2

{ }限定了内嵌if范围,{ }内是一个整体。

\*The { } limits the scope of the embedded if, and the { } is a whole.

switch语句实现多分支选择结构  
Implementation of Multi-Branch Selection Structure by Switch Statement

例4.5 编写程序，输入成绩的等级，输出分数。

Example 4.5 Programming, input grade, output score.

学生成绩分类

Classification of Students'Achievements

85～100 points are 'A'

70～84 points are 'B'

60～69 points are 'C'

<60 points are `D'

#include <stdio.h>

int main()

{ char grade;

scanf("%c",&grade);

switch(grade)

{ case 'A': printf("Your score:85～100\n");break;

case 'B': printf("Your score:70～84\n");break;

case 'C': printf("Your score:60～69\n");break;

case 'D': printf("Your score:<60\n");break;

default: printf("enter data error!\n");

}

return 0;

}

If deleted break:

scanf("%c",&grade);

printf("Your score:");

switch(grade)

{ case 'A': printf("85～100\n");

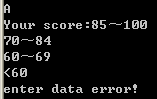
case 'B': printf("70～84\n");

case 'C': printf("60～69\n");

case 'D': printf("<60\n");

default: printf("enter data error!\n");

}



\*Enter from a, execute all the way to “}”

1. **循环结构程序设计**

C语言提供三种循环语句

C language provides three kinds of loop statements

for

while

do……while

用for 语句实现循环(Loop with if statement)

for语句的一般形式(General form)：

for(表达式1；表达式2；表达式3)

语句

for 语句的执行过程

The execution process of the for statement

step1:Calculation expression 1

step2:Judge expression 2, if it is true, execute the statement,then goto step 3, otherwise goto step 5

step3:Calculation expression 3

step4:Judge expression 2, if it is true, execute the statement,then goto step 3, otherwise goto step 5

step5:end the loop

例： 求1×2×3×4×5

#include <stdio.h>

int main()

{

int i,p=1;

for(i=2;i<=5;i++)

p=p\*i;

printf(“p=%d\n",p);

return 0;

}

若求1+1/3+1/5+1/7+1/9+1/11

#include <stdio.h>

int main()

{

int i,p=1;

for(i=3;i<=11;i=i+2)

p=p+1/i;

printf("p=%d\n",p);

return 0;

}

用while语句实现循环(Loop with while statement)

while语句的形式如下(General form)：

while (表达式)

语句

while循环的特点是：

先判断条件表达式，后执行循环体语句

The characteristics of the while :

Judge the condition expression first, and then execute the loop body statement.

例5.3求1+2+3+…+100

#include <stdio.h>

int main()

{

int i=1,sum=0;

while (i<=100)

{

sum=sum+i;

i++;

}

printf("sum=%d\n",sum);

return 0;

}

例：输入一批学生的成绩（以负数结束），计算并输出最高分

\*\*\*Example : Enter the scores of a group of students (end with a negative number), calculate and output the highest score

……

max=0;

scanf("%d",&grade);

while(grade>=0)

{

if(grade>max)

max=grade;

scanf("%d",&grade);

}

printf("The max grade is %d\n",max);

……

用do---while语句实现循环(Loop with do…while statement)

do---while语句的形式如下(General form)：

do

{

语句；

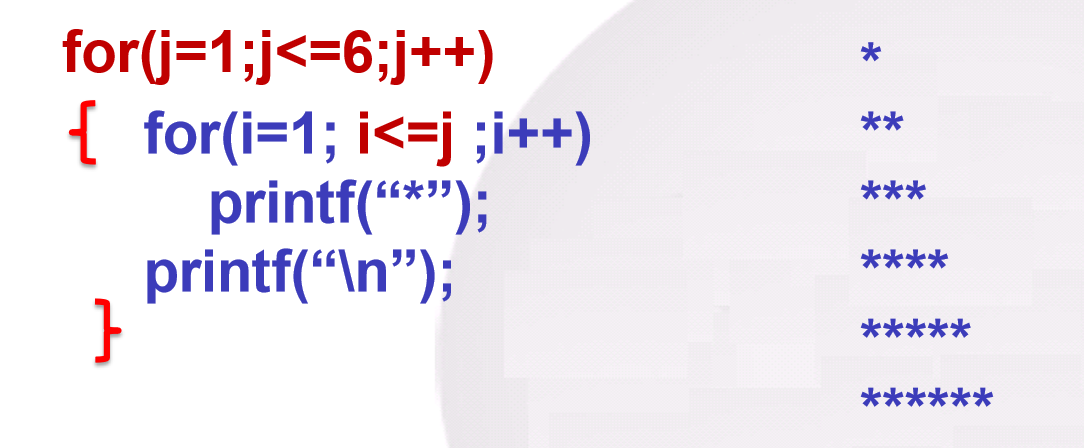
} while (表达式)；

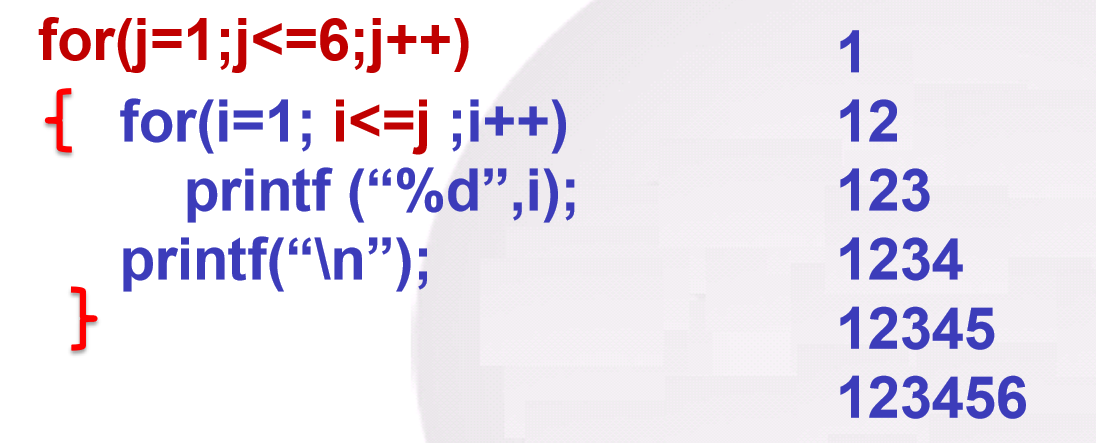
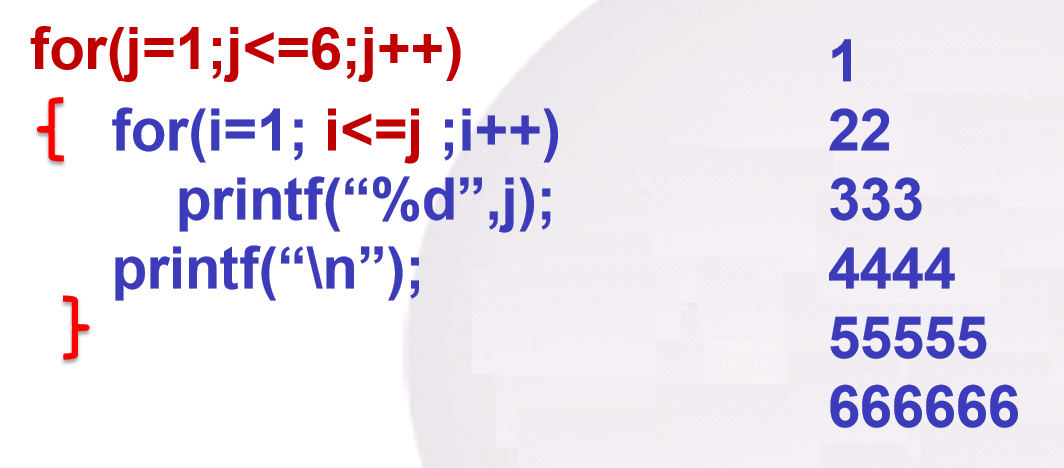
do---while的特点：先执行循环体，然后判断循环条件是否成立

Characteristic: first execute the loop body, and then judge whether the loop condition is true or not.

编程使用循环输出以下图案：

Program output the following patterns using the loop:





改变循环执行的状态  
(Change the state of loop execution)

用break语句提前终止整个循环

Use the break statement to end entire loop early

用continue语句提前结束某一次循环

Use the continue statement to end the loop early

break语句可以用来从循环体内跳出循环体，即提前结束循环，接着执行循环下面的语句。

The break statement can be used to jump out of the loop body, that is, to end the loop ahead of time, and then execute the statement below the loop.

有时并不希望终止整个循环的操作，而只希望提前结束本次循环，而接着执行下一次循环。这时可以用continue语句

Sometimes you do not want to terminate the operation of the entire loop, but only want to end the loop ahead of time, and then execute the next loop. You can use the continue statement

循环结构程序设计举例  
examples of loop structure programming

找最大值或最小值  
Find the maximum or minimum value

例\*\*5.9 输入10个整数，编程求其中的最大值。

Example 5.9 input 10 integers and program to get the maximum value.

#include <stdio.h>

int main()

{ int n,max,i; //max stores the maximum value, i is the loop variable

scanf("%d",&n); //Enter the first number

max=n; //Default the first number to the maximum

for(i=1;i<=9;i++)

//Read the following 9 numbers in sequence for comparison

{

scanf("%d",&n);

if(n>max) //If the number read in is greater than the maximum,

max=n; //use it as the maximum

}

printf("max=%d\n",max);

return 0;

}

枚举法求解问题

Solving problems by enumeration method

编程输出100-200之间的所有的奇数（不能被2整除的数称为奇数）（要求每行输出5个数，并用2个空格间隔开）。

Program to output all odd numbers between 100-200 (numbers that cannot be divided by 2 are called odd numbers) (requires 5 numbers to be output per line and separated by 2 spaces).

#include <stdio.h>

int main()

{ int i,count=0; //count is used to count

for(i=100;i<=200;i++)

{ if(i%2!=0)

{

printf("%-5d",i);

count++;

if(count%5==0)

printf("\n");

}

}

return 0;

}

1. **数组**

**一组同类型数据的集合称为数组（Array），它是一组有序数据的集合。**

**A set of data of the same type is called an array, which is a set of ordered data.**

**它所包含的每一个数据叫做数组元素（Element），所包含的数据的个数称为数组长度（Length）。**

**Each data it contains is called an element, and the number of data it contains is called the length of the array.**

**定义float类型的数组s，该数组有10个元素**

**Define array s of float type, which has 10 elements**

**float s[10];**

**用数组名和下标惟一确定数组中的元素。**

**The array name and subscript are used to uniquely determine the elements in the array.**

**如s[0]、 s[1]、 s[2]…… s[9]**

**一个数组元素相当于一个变量**

**an array element is equivalent to a variable**

**数组中的每一个元素都属于同一种数据类型。不能把不同类型的数据（如学生的成绩和学生的性别）放在一个数组中。**

**Each element in the array belongs to the same data type. You cannot put different types of data (such as student grades and student gender) in one array.**

**一维数组(One-dimensional array)**

**数组必须先定义后使用**

**Array must be defined before use**

**一个数组元素相当于一个变量**

**an array element is equivalent to a variable**

**注意：数组元素的下标从0开始**

**Note: the subscript of array element starts from 0**

**一般形式(General form)：**

**类型符 数组名[常量表达式];**

**如 int a[10];**

**常量表达式可以是常量和符号常量。**

**Constant expressions can be constants and symbolic constants.**

**int s[5+5];**

**#define N 10**

**int s[N];**

**常量表达式不能是变量。**

**Constant expressions cannot be variables**

**int n=10; int s[n]; （wrong）**

**一维数组的初始化  
Initialization of one dimensional array**

**int a[5]={1,2,3,4,5};**

**If the number of data is the same as the length of the array, the length of the array can be omitted**

**int a[5]={1,2,3,4,5};**

**int a[ ]={1,2,3,4,5};**

**int b[5]={1,2,3,0,0};**

**int b[5]={1,2,3};**

**int c[5 ]={0,0,0,0,0};**

**int c[5]={0};**

**注意(Notice)：**

**如果不进行初始化，如定义int d[5]，那么数组元素的值是随机的，系统不会将其设置为0。**

**\*If you do not initialize, such as defining int d [5], the values of array elements are random and will not be set to 0 by system.**

**以下定义是错误的，即给定初值的个数不能大于数组定义的长度。**

**The following definition is wrong, that is, the number of given initial values cannot be greater than the length of the array definition.**

**int a[3]={1,2,3,4,5};（wrong）**

**一维数组元素的引用  
References to one dimensional array elements**

**数组名［下标］**

**如\*s[0]=s[5]+s[7]-s[2\*3]**

**int n=5,a[10];**

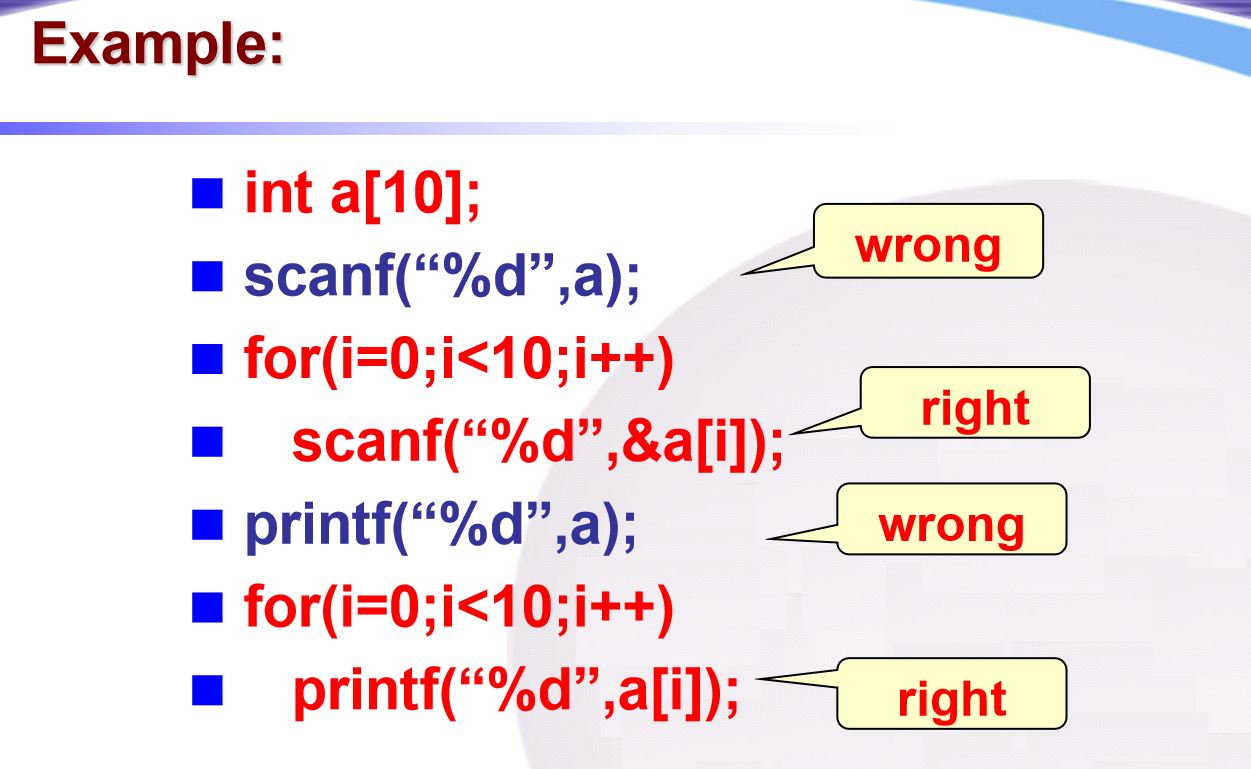
**a[n]=20; 相当于 a[5]=20;**

**a ['c'-'a']=5;相当于 a[2]= 5;**

**a[10]=20;（wrong,下标越界（0-9）subscript out of range）**

**注意：只能引用数组元素而不能一次整体调用整个数组**

**Note: you can only reference array elements and not use the whole array at one time**



例6.3 对10个数组元素依次赋值为0,1,

2,3,4,5,6,7,8,9，要求按逆序输出。

Example 6.3 assign 0,1,2,3,4,5,6,7,8,9,to 10 array elements. it is required to output in reverse order.

#include <stdio.h>

int main()

{ int i,a[10];

for (i=0; i<=9;i++)

a[i]=i;

for(i=9;i>=0; i--)

printf("%d ",a[i]);

printf("\n");

return 0;

}

例\*\*\*6.4 任意输入10个数，找出其中的最大值和最小值。

Example 6.4 input 10 numbers randomly to find out the maximum and minimum values.

#include <stdio.h>

#define N 10 //Symbolic constant

int main()

{ int i,a[N],max,min;

for (i=0; i<N;i++) //Use loop to input 10 numbers

scanf("%d",&a[i]);

max=min=a[0];

for (i=1; i<N;i++)

{

if(a[i]>max) max=a[i];

if(a[i]<min) min=a[i];

}

printf("max=%d\n",max);

printf("min=%d\n",min);

return 0;

}

例6.5 利用顺序查找的原理判断在数组中是否包含指定值。

Example 6.5 use the principle of sequential search to determine whether the specified value is included in the array.

#include<stdio.h>

int main()

{

int a[10]={67,87,32,4,6,8,9,3,23,56};

int x,temp=0,i;

printf("请输入要查找的值：");

scanf("%d",&x);

for(i=0;i<10;i++)

{ if(a[i]==x)

{ temp=1; //temp as a flag variable

}

}

if(temp==0)

printf("查找失败！");

else

printf("查找成功！");

return 0;

}

二维数组的定义  
Definition of two dimensional array

一般形式(General form):

类型符 数组名[常量表达式][常量表达式];

如： int a[3][4];

float b[5][10];

二维数组的初始化  
Initialization of Two-dimensional array

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

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

int a[2][3]={{1,3,5},{0,0,0}};

int a[2][3]={{1,3,5}};

int a[2][3]={{1,0,0},{2,0,0}};

int a[2][3]={{1},{2}};

int a[3][2]={{1,0},{0,0},{7,0}};

int a[3][2]={{1},{0},{7}};

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

等价于(Equivalent to):

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

二维数组元素的引用  
References to two dimensional array elements

形式(form)：

数组名［下标］［下标］

二维数组程序举例  
Two dimensional array program example

例6.9 有一个3×4的矩阵，要求编程序求出其中值最大的那个元素的值，以及其所在的行号和列号。

Example 6.9 has a 3 × 4 matrix, which requires the program to find out the value of the element with the largest value, as well as the row number and column number.

……

int i,j,row,colum,max,a[3][4];

for (i=0;i<=2;i++) //intput array a

for (j=0;j<=3;j++)

scanf("%d",&a[i][j]);

max=a[0][0]; row=0;colum=0;

for (i=0;i<=2;i++)

for (j=0;j<=3;j++)

if (a[i][j]>max)

{ max=a[i][j]; row=i; colum=j; }

printf("max=%d\nrow=%d\ncolum=%d\n",max,row,colum);

……

字符数组Character array

用来存放字符数据的数组是字符数组

The array used to store character data is called character array

字符数组中的一个元素存放一个字符

An element in a character array holds a character

char c[10];

char c[10]={’I’,’ ’,’a’,’m’,’ ’,’h’,’a’,’p’,’p’,’y’};

char diamond[5][5]={{’ ’,’ ’,’\*’},

{’ ’,’\*’,’ ’,’\*’},

{’\*’,’ ’,’ ’,’ ’,’\*’},

{’ ’,’\*’,’ ’,’\*’},

{’ ’,’ ’,’\*’} };

字符串结束标志String end flag

关心的是字符串的有效长度而不是字符数组的长度

It's about the effective length of the string, not the length of the character array

为了测定字符串的实际长度，C语言规定了字符串结束标志’\0’

In order to determine the actual length of a string, C specifies the string end flag '\0'

char c[]={”I am happy”};

char c[]=”I am happy”;

char c[11]=”I am happy”;

char c[10]={”China”};

char c[10]=”China”;

从c[5]开始，元素值均为\0

From C [5], the element values are all \0

字符数组的输入输出  
Input and output of character array

用%s输出字符串时，printf函数中的输出项是字符数组名，不是数组元素名。

When you output a string with %s, the output item in the printf function is a character array name, not an array element name.

char a[10]=“china”;

printf(“%s”,a);

scanf函数中的输入项是已定义的字符数组名（代表数组的首地址），输入的字符串应短于已定义的字符数组的长度

The input item in the scanf function is the name of the defined character array (representing the first address of the array), and the input string should be shorter than the length of the defined character array

char c[6];

scanf(” %s ”,c); //数组名array name

字符串处理函数String processing function

7. strcmp

一般形式(General form):

strcmp(字符串1，字符串2)

作用是比较字符串1和字符串2的大小

To compare the big or small of strings 1 and 2

比较的结果由函数值带回

The result of the comparison is brought back by the function value

如果字符串1等于字符串2，则函数返回值为0

If string 1 equals string 2, the return value of the function is 0

如果字符串1大于字符串2，则函数返回一个正整数

If string 1 is greater than string 2, the function returns a positive integer

如果字符串1小于字符串2，则函数返回一个负整数

If string 1 is less than string 2, the function returns a negative integer

char str1[10]=“Chop”,str2[]=”China”;

if(str1>str2) printf(”yes”);

else printf(”no”); (wrong)

if( strcmp(str1,str2)>0)

printf(”yes”);

else printf(”no”); (correct)

考试形式：在线闭卷 Exam format: Online closed book

考试平台：智慧树 Exam platform: Smart Tree

考试时长：120分钟 Exam duration: 120 minutes

考试时间：12月26日15：00-17：00

Exam time: December 26th, 15:00-17:00

试卷题型Exam question types：

1.单选题（20小题，共40分）Single choice question (20 questions, 40 points in total)

2.判断题（10小题，共20分）True or False (10 questions, 20 points in total)

3.代码填空题（根据程序功能填写部分代码行）（5个空行，共10分）Code fill in the blank question (fill in some lines of code according to program functions) (5 blank lines, 10 points in total)

4.修改代码题（根据程序功能修改代码行）（5个错误行，共10分）Revise code questions (modify code lines based on program functionality) (5 error lines, total 10 points)

5.编程题（根据题目要求编写完整源代码）（2小题，共20分）Programming Question (Write complete source code according to the requirements of the question) (2 questions, total 20 points)

1. 输入x、y两个整数，按从大到小的顺序输出x、y。input two integers, x and y, and output them in descending order.
2. 从键盘上输入若干学生的成绩，统计并输出最高成绩和最低成绩，当输入负数时结束输入。input the grades of several students from the keyboard, calculate and output the highest and lowest grades, and end the input when a negative number is entered.
3. 判断输入的字符是否为大写字母，如果是大写字母，将其转换为小写字母，否则就不转换，最后输出。determine whether the input characters are uppercase letters. If they are uppercase letters, convert them to lowercase letters. Otherwise, do not convert them, and finally output.
4. 输入10个整数存放到数组中，然后再输入1个数，查找该数是否在数组中。input 10 integers and store them in an array, and then input 1 number to check if the number is in the array.
5. 编写程序打印以下图案（用循环的嵌套实现）Write a program to print the following pattern (implemented using nested loops)。

\*

\*\*

\*\*\*

\*\*\*\*

1. 从键盘输入一个正整数x，判断它是否为3和7的倍数，若是，输出yes，否则输出no。Write a program to implement the following function: input a positive integer x from the keyboard, determine if it is a multiple of 3 and 7, if so, output yes, otherwise output no.