# 第四次上机作业

目录

[第四次上机作业 1](#_Toc402215529)

[代码风格要求：利用“TAB”键进行合理的缩进以强调语句间的从属关系 2](#_Toc402215530)

[work1：if-else结构练习：求数的绝对值 4](#_Toc402215531)

[work2：if-else结构练习：求三角形面积 4](#_Toc402215532)

[work3：if-else if-else结构练习：运输问题 4](#_Toc402215533)

[work4：switch-case结构练习：运输问题 4](#_Toc402215534)

[work5：三种循环结构基本语法练习：数的遍历 4](#_Toc402215535)

[work6：三种循环结构基本语法练习：数的条件遍历 4](#_Toc402215536)

[work7：VS中的断点和单步调试技术 4](#_Toc402215537)

[work8：VS中的断点和单步调试技术 5](#_Toc402215538)

[work9：简单循环练习：数的条件遍历 6](#_Toc402215539)

[work10：简单循环练习：累积 6](#_Toc402215540)

[work11：简单循环练习：数的简单处理 6](#_Toc402215541)

[work12：简单循环练习：数的条件遍历 6](#_Toc402215542)

代码风格要求：利用“TAB”键进行合理的缩进以强调语句间的从属关系

参考：**高质量C/C++编程.pdf**：

1. if(表达式)

{

语句1;

语句2;

}

1. if(表达式)

{

多条语句;

}

else if(表达式)

{

多条语句;

}

else

{

}

1. switch(表达式)

{

case x:

语句段;

break;

case y:

语句段;

break;

default:

语句段

break;

}

1. while(表达式)

{

语句1;

语句2；

}

1. do

{

语句1;

语句2;

}while(表达式);

1. for(表达式1,表达式2,表达式3)

{

语句1;

语句2;

}

1. main函数中：

int main(void)

{

while(表达式)

{

语句1;

语句2;

}

return 0;

}

work1：if-else结构练习：求数的绝对值

熟悉if-else结构，要求用户从键盘输入一个实数，编程输出该数的绝对值（不能使用math中的绝对值函数）。

work2：if-else结构练习：求三角形面积

熟悉if-else结构，要求用户从键盘输入三角形的三条边a,b,c，如果用户输入的三边不能构成三角形，则提示用户输入失败；否则，输出该三角形面积。

work3：if-else if-else结构练习：运输问题

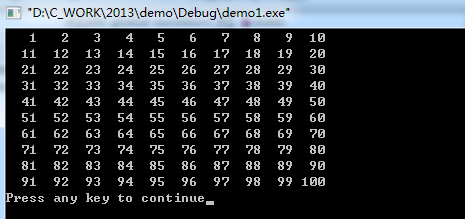
利用if-else if-else结构，实现Chapter4-Switch.pdf中最后一页的运输问题。

work4：switch-case结构练习：运输问题

利用switch-case结构，实现Chapter4-Switch.pdf中最后一页的运输问题。考虑：switch-case语句的基本语法组成；switch-case结构和if-else if-else结构有何异同？

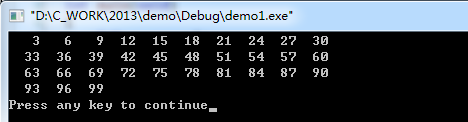
work5：三种循环结构基本语法练习：数的遍历

分别利用while/do-while/for循环结构，打印从1到100的数，每个数占四列，每行输出10个数。作业标号分别work5\_while.c work5\_do\_while.c work5\_for.c。注意三种循环结构的基本语法。



work6：三种循环结构基本语法练习：数的条件遍历

在work5的基础上，输出1-100内能够被3整除的数，每个数占四列，每行输出10个。Work5\_while.c work5\_do\_while.c work5\_for.c



work7：VS中的断点和单步调试技术

下列代码试图计算，但是程序中存在若干处逻辑错误，请利用VS提供的断点和单步跟踪技术，找到并改正相应逻辑错误，使得程序能够得到正确的计算结果。调试方法参考群共享中的视频。需要掌握的调试技巧：在特定代码处增加断点（F9）；控制代码单步执行（F10）；设置观察变量，将变量变化的情况，与自己的执行逻辑进行对比，判断计算机是否是按照程序员的意图在执行指令，如果没有，那么代码在哪里引发了错误？**必须一行一行的逐行进行判断**。提示：观察i值和sum值的变化。

VS中与调试相关的若干快捷键：

1. 增加断点：将光标移动到希望停止的代码行，按F9
2. 开始调试：F5
3. 单步执行：F10
4. 终止调试：Shift + F5

#include <stdio.h>

int main(void)

{

double sum;

int i;

while(i < 10)

{

sum += 1.0 / i;

}

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

return 0;

}

work8：VS中的断点和单步调试技术

下列代码试图从键盘获取一个十进制整型数，并输出该数各个位上的数字乘积，如输入为26时，输出为2\*6=12，但是程序中存在若干处逻辑错误，请利用VS提供的断点和单步跟踪技术，找到并改正相应逻辑错误，使得程序能够得到正确的计算结果。提示：观察k值和num值的变化情况是否符合程序员的意图。

#include <stdio.h>

int main(void)

{

int num;

int k;

printf("please input a number:\n");

scanf("%f",&num);

do

{

k \*= num % 10;

num /= 10;

}while(num == 0);

printf("%d\n",k);

return 0;

}

work9：简单循环练习：数的条件遍历

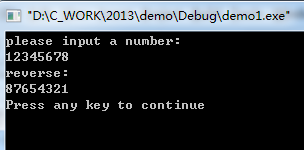
编写循环程序，打印1900-2013年中所有的闰年。提示：遍历1900-2013中的每一个整数，判断其是否满足闰年的条件。

work10：简单循环练习：累积

编写循环程序，计算，其中x和y均由键盘输入。提示：循环次数由y确定，循环体为累积（考虑一下我们是如何实现累加的？）

work11：简单循环练习：数的简单处理

输入一个任意长的整数，将其逆序输出。



work12：简单循环练习：数的条件遍历

求所有具有性质的四位数，如，提示：遍历1000-10000中的每个整数。

