```
 1   2   3   4   6   7   8   9   10
Used continue to skip printing the value : 5
```

例 3-29　编写一个程序，功能是打印出 1～100 之间不能被 3 整除的数，并要求输出结果时，10 个数一行。

解　（1）程序实现

```
#include "stdio.h"
main()
{
  int x = 0, y = 0;
  while (++x <= 100)
  {
    if (x % 3 == 0)
      continue;
    printf ("%4d", x);
    ++y;
    if (y %10 == 0)
      printf ("\n");
  }
  printf ("\n");
}
```

（2）分析与讨论

该程序主要由一个 while 循环构成。由于希望打印出 1～100 间不被 3 整除的数，所以在循环体里判断出某个数能够被 3 整除时，就用 continue 结束此次循环而进入下一次循环。程序中，把修改循环控制条件的语句，与判断继续循环的条件合二为一，成为：

```
++x <= 100
```

另外，程序中用变量 y 来控制每行打印 10 个数据。图 3-21 所示为该程序运行的结果。

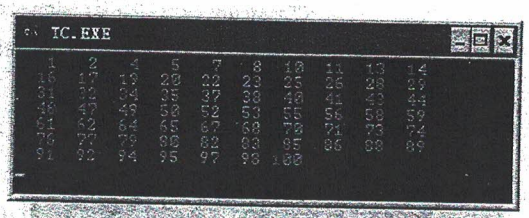

图 3-21　例 3-29 的运行结果

3.3.5　循环的嵌套结构

如果在一个循环结构的循环体内，又出现了一个循环结构，那么这就是所谓的“循环的嵌套结构”，有的书上称其为多重循环。既然是嵌套式的结构，那就表明各循环之间只能是“包含”关系，即一个循环结构完全在另一个循环结构的里面。通常把里面的循环称为“内循环”，外面的循环称为“外循环”。

C 语言的 3 种循环语句都可以嵌套，既可以自身嵌套，也可以相互嵌套。比如，while 语句可以出现在 for 语句的循环体里，for 语句也可以出现在 do…while 语句的循环体里，如此等等。另外，循环嵌套的层数没有限制，但一般用得较多的是二重循环或三重循环。

比如，有如下的循环嵌套语句片段：

```
int i, j, nr = 0;
for (i = 1; i<100; i++)              /* 外循环 */
{
  for (j = i; j<=100; j++)                /* 内循环 */
  {
    nr = nr + 1;
  }
}
```

试问语句“nr = nr + 1;”总共执行多少次？

由于内循环在外循环的循环体里，所以内循环这个整体将被执行 99 次（因为控制外循环的变量 i 是从 1 变到小于 100，一共 99 次）。而内循环循环体本身（即语句 nr = nr + 1;）每次要执行的次数是多少呢？从内循环 for 中的几个表达式可以看出，它每次的执行次数与外循环变量 i 的当前取值有关，是一个不定的数。当内循环第 1 次被执行时，它的循环控制变量 j 的初始值为 1，所以它的循环体将执行 100 次；当内循环第 2 次被执行时，它的循环控制变量 j 的初始值为 2，所以它的循环体将执行 99 次；……；当内循环最后一次（即第 99 次）被执行时，它的循环控制变量 j 的初始值为 99，所以它的循环体将执行 2 次。至此可知，内循环体总共被执行：100+99+…+2=5049 次。这就是语句“nr = nr + 1;”总共执行多少次的答案。

为了在编程时明确语句结构间的各种关系，一般都采用“缩进”的格式书写程序。其实，我们在前面编程时，一直都采用着这种缩进的格式，这是一种良好的书写程序的习惯。

例 3-30 编写一个程序，能够输出如图 3-22 中显示的表格形式，即打印出 1～9 这 9 个数的 2，3，4 倍数。

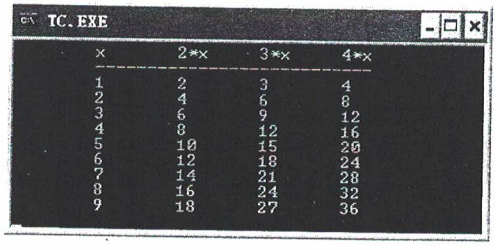

图 3-22 1～9 这 9 个数的 2，3，4 倍数

解 （1）程序实现

```
#include "stdio.h"
main()
{
  int j, k, x;
```

```
    printf ("\tx\t2*x\t3*x\t4*x\n");
    printf ("\t----------\n");
    for (j =1; j<=9; j++)                    /* 外循环开始 */
    {
      for (k = 1; k<=4; k++)                 /* 内循环开始 */
      {
        x = j*k;
        printf ("\t%d", x);
      }                                      /* 内循环结束 */
      printf ("\n");                         /* 进入下一行打印 */
    }                                        /* 外循环结束 */
  }
```

（2）分析与讨论

从图上看出，这可以是一个两重循环问题：外循环是从 1 做到 9，控制表格行的输出；内循环是控制每行 4 个数据的输出，即循环 4 次。这里要注意，由于打印一行后，就应进入下一行打印，所以在内循环结束、还没有进入下一次外循环时，要输出一个回车换行，即做：printf ("\n");。

例 3-31　不断地从键盘输入两个正整数，求它们的最大公约数。直到用户回答“n”时，停止程序的运行。

解　（1）程序实现

```
#include "stdio.h"
main()
{
  int x1, x2;
  char ch;
  while (1)
  {
     printf ("Please enter two positive integers:");
     scanf ("%d%d", &x1, &x2);
     getchar();              /* 让过 scanf 最后的回车符 */
     do
     {
       if (x1>x2)
          x1 -= x2;
       else if (x2>x1)
          x2 -= x1;
     }while (x1 != x2);
     printf ("The greatest common divisor is %d\n", x1);
     printf ("Do you want to continue?(y or n)");
     ch=getchar ();
```

```
        getchar();              /* 让过前面 getchar 的回车符 */
        if (ch == 'n')
          break;
      }
    }
```

（2）分析与讨论

整个程序由两重循环构成：while 是外循环，do…while 是内循环。在 while 循环语句的圆括号里，1 表示条件永远为真，即循环一直要做下去。正因为这样，在进入 while 循环之前，并没有为它准备循环控制条件的初值语句。该外循环只有当在它的循环体最后往变量 ch 里输入了一个字符“n”时，才通过 break 语句强制结束循环。

do…while 内循环是通过辗转相减的方法求两个正整数的最大公约数的。比如有两个正整数为 x1 和 x2，那么辗转相减求两个正整数的最大公约数的步骤如下。

第 1 步：如果 x1>x2，则 x1=x1−x2；

第 2 步：如果 x2>x1，则 x2=x2−x1；

第 3 步：如果 x1==x2，则算法结束，当前 x1（或 x2）的值就是所求的最大公约数；否则重复上述步骤。

图 3-23 所示为程序循环执行 3 次的情形：第 1 次输入的两个数为 30 和 192，最大公约数是 6；第 2 次输入的两个数为 152 和 44，最大公约数是 4；第 3 次输入的两个数为 55 和 78，最大公约数是 1。最后由于键入了字母 n，于是通过 break 使循环强行结束。

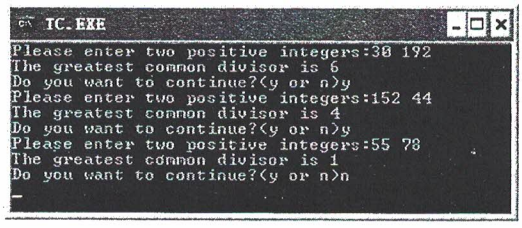

图 3-23　3 次求最大公约数的执行情况

例 3-32　用嵌套循环来改写例 3-22 的求 100～999 之间所有的水仙花数。

解　（1）程序实现

```
#include "stdio.h"
main()
{
  int x, y, nf, ns, nt, count = 0;
  for (nt = 1; nt <= 9; nt ++)
    for (ns = 0; ns <= 9; ns ++)
      for (nf = 0; nf <= 9; nf ++)
      {
        x = nt*100 + ns*10 + nf;
        y = nf*nf*nf + ns*ns*ns + nt*nt*nt;
        if (x == y)
```

```
        {
            printf ("The %d' narcissus number is : %d\n", count+1, x);
            count++;
        }
    }
}
```

（2）分析与讨论

整个程序是 for 的 3 重循环，形成从 100 开始到 999 之间的所有正整数。在程序中，变量 nf 里是个位数字，ns 里是十位数字，nt 里是百位数字。通过循环，这些数字在变量 x 里形成一个个介于 100～999 之间的三位数，在变量 y 里形成它们的立方和。这样，如果条件 x == y 成立，那么它肯定就是一个水仙花数。于是，一方面将当前的 x 值打印出来，另一方面由变量 count 进行计数。

习　题　3

一、填空

1．若变量 x、y、z 都是 int 型的。现有语句：

```
scanf ("%3d%4d%2d", &x, &y, &z);
```

假定在键盘上输入 123456789↙。那么变量 x 里是_____，y 里是_____，z 里是_____。

2．若变量 x、y、z 都是 int 型的。现有语句：

```
scanf ("%d,%d,%d", &x, &y, &z);
```

为了使 x 里是 12，y 里是 345，z 里是 187，应该在键盘上键入_____。

3．程序填空。

```
#include"stdio.h"
main()
{
  int x, y;
  scanf ("%d", &x);
  y=x%2;
  switch( ① )
  {
    case 0:
      printf ("It is a even integer.\n");
      ②;
    default:
      printf ("It is a odd integer.\n");
  }
}
```

4．循环：for (x=0; x != 123;) scanf ("%d", &x); 在_____时被终止。