循环中的 continue 语句

让我们想一想下面程序的输出结果是什么?

```
#include<iostream>
using namespace std;
int main()
{
    int a = 10;
    do
    {
        if(a == 15)
        {
            a = a + 1;
            continue;
         }
         cout << a << endl;
         a = a + 1;
    }while( a < 20 );
    return 0;
}
```

这就是程序的输出结果,实际上就是在屏幕上输出了除了 15 以外,从 10 到 19 的数字。

```
10
11
12
13
14
16
17
18
19
```

C++中的 continue 语句有点像 break 语句。但它不是强迫终止整个循环，而是会跳过当前循环中的代码，强迫开始下一次循环。

对于 for 循环，continue 语句会执行条件测试和循环增量部分。对于 while 和 do-while 循环，continue 语句会回到条件测试上。

循环中的 continue 语句

让我们想一想下面程序的输出结果是什么?

```
#include<iostream>
using namespace std;
int main()
{
    int a = 10;
    do
    {
        if(a == 15)
        {
            a = a + 1;
            continue;
         }
         cout << a << endl;
         a = a + 1;
    }while( a < 20 );
    return 0;
}
```

这就是程序的输出结果,实际上就是在屏幕上输出了除了 15 以外,从 10 到 19 的数字。

```
10
11
12
13
14
16
17
18
19
```

C++中的 continue 语句有点像 break 语句。但它不是强迫终止整个循环，而是会跳过当前循环中的代码，强迫开始下一次循环。

对于 for 循环，continue 语句会执行条件测试和循环增量部分。对于 while 和 do-while 循环，continue 语句会回到条件测试上。

逢 7 过

【问题描述】

小智和朋友在玩逢 7 过的游戏，游戏规则：游戏开始者报小于 100 的正数，按照顺时针每个人开始报数，将要报的数中如果包含 7 或者是 7 的倍数，报数人应说“过”，如果出错，将接受游戏的惩罚，游戏结束。如果到达 100 没人出错，游戏也结束。小智现在想通过程序找出所有能报的数字，你能帮他实现一下吗？

【输入样例】

```
45
```

【输出样例】

```
45 46 48 50 51 52 53 54 55 58 59 60 61 62 64 65 66 68 69 80 81 82
83 85 86 88 89 90 92 93 94 95 96 99 100
```

【参考代码】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for(int i = n; i <= 100; i++)
    {
        if (i % 7 == 0)
        {
            continue;
        }

        if (i / 10 == 7)
        {
            continue;
        }
        if (i % 10 == 7)
        {
            continue;
        }

        cout << i << ' ';
```

```
    }
    return 0;
}
```

电子停车计费器

【问题描述】

小智在做一个电子停车计费器，每天停车位上都有若干车辆停泊，今天记录了一个停车位上的停车时间，请统计一下这个车位有多少次收费停车，各收多少钱，以及收费的总额。

停车时间在 15 分钟以内的不收费，然后每 15 分钟收费 1 元钱，根据真实时长按比例收费。

输出金额保留 2 位有效数字，数据之间用一个空格隔开。

【输入格式】

有 2 行，第一行是停车记录次数，

第二行是每次停车时长（分钟）。

【输出格式】

有若干行，每一行包括收费流水号，停车时长，停车费用。

最后一行是停车费总额。

【输入样例】

```
10
20 12 87 33 2 46 100 23 19 58
```

【输出样例】

```
1 20 1.33
2 87 5.80
3 33 2.20
4 46 3.07
5 100 6.67
6 23 1.53
7 19 1.27
8 58 3.87
25.73
```

【算法分析】

在循环中依次输入每次停车的时长。

对于低于 15 分钟的停车不计费，跳过循环。

累加每次停车的费用。

输出每次收费流水号，停车时长和费用。

最后输出费用累加值。

【参考代码】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double sum = 0; //总额
    int sn = 1; //流水号
    int n, x; //n 次停车；每次时长 x（分钟）
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        cin >> x; //录入停车时长
        if(x < 15) //不满 15 分钟不计费
        {
            continue;
        }
        double fee = x / 15.0; //计算停车费
        cout << sn << " " << x << " "; //输出流水号和时长
        cout << fixed << setprecision(2) << fee << endl; //单次停车费
        sum += fee;
        sn += 1;
    }
    cout << fixed << setprecision(2) << sum << endl; //停车费总额
    return 0;
}
```

循环语句大师挑战

【问题描述】

小智的朋友会使用二重循环来输出一个直角三角形的图案，小智不服气，和他朋友进行了一场循环语句大师挑战。

小智如果用一重循环就能做到同样的事情，他的朋友就尊称小智为循环语句大师。

你来帮小智完成这项挑战吧!

【输入格式】

有 1 行，一个正整数 n（n <= 20)。

【输出格式】

有 n 行，第 i 行有 i 个星号，形成一个直角三角形的图案。

【输入样例】

```
7
```

【输出样例】

```
*
**
***
****
*****
******
*******
```

【算法分析】

需要记录每行的行号，以及每行的星号数量。

在循环中，当一行中输出的星号数量小于行号数值时，输出星号，增加星号的数量，然后跳过后面的循环，不进行换行。

当星号数量达到了行号数值时，换行，行号增加。

【参考代码】

```
#include<iostream>
```

```
using namespace std;
int main()
{
    int n;
    cin >> n;//三角形图案的高度
    int stars = 0;//星号数量
    for (int line = 1; line <= n; )
    {
        //星号数量不够时
        if(stars < line)
        {
            //一直输出星号
            cout << '*' ;
            stars++;
            continue;
        }
        if (i % 10 == 7)
        {
            continue;
        }

        cout << i << ' ';
    }
    return 0;
}
```