

集训课-06：基础数论1

课程目标

01

break与continue

02

约数

03

质数

04

cmath模块

05

总结回顾



Part 1

**break与
continue**

break与 continue

break;

在程序中表示结束循环

break与 continue

输出1~4，然后结束程序

```
for (int i = 1; i <= 6; i++)  
{  
    cout << i << " ";  
    if (i == 4)  
    {  
        break;  
    }  
}
```

break与 continue

`continue;`

在程序中表示跳过本次循环，继续下一次循环

break与 continue

输出1~3，跳过4，接着输出5~6

```
for (int i = 1; i <= 6; i++)  
{  
    if (i == 4)  
    {  
        continue;  
    }  
    cout << i << " ";  
}
```



Part 2

约数

约数

约数：

整数 a 除以整数 b ($b \neq 0$) 除得的商正好是整数而没有余数

b 称为 a 的约数

约数

下面哪些数是6的约数？

6

1

2

3

4

5

约数

程序中判断是否是6的约数：

```
if (6 % 3 == 0)           //结果为true，3是6的约  
数
```

```
if (6 % 2 == 0)           //结果为true，2是6的约  
数
```

```
if (6 % 4 == 0)           //结果为false，4不是6的约  
数
```

约数

一个数最小的约数：1

一个数最大的约数：它本身

约数

如何输出一个数所有的约数？

约数

输出一个数所有的约数

Step1: for循环从1到这个数本身

Step2: 判断1-它本身之间的数是不是它的约数

Step3: 如果是, 则输出

约数

输出一个数所有的约数

求
n
所有的
约数

```
int n;  
cin >> n;  
for (int i = 1; i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        cout << i << " ";  
    }  
}
```

for
循环和
if 嵌套

约数

如何输出一个数的约数个数？

约数

输出一个数的约数个数

Step1：创建计数器变量 $cnt = 0$

Step2：for循环从1到这个数本身

Step3：判断1-它本身之间的数如果是约数，则把cnt变量增加1

Step4：输出计数器变量cnt

约数

输出一个数的约数个数

求 n 所有的约数个数

```
int n, cnt = 0;  
cin >> n;  
for (int i = 1; i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        cnt++;  
    }  
}  
cout << cnt;
```

过程性：统计

结果性：输出

for 循环和 if 嵌套



Part 3

质数

质数

质数：

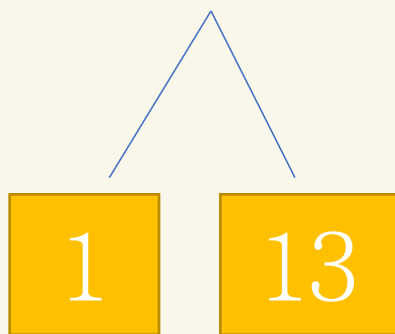
一个自然数只能被1和它本身整除的数

注意：1不是质数

质数

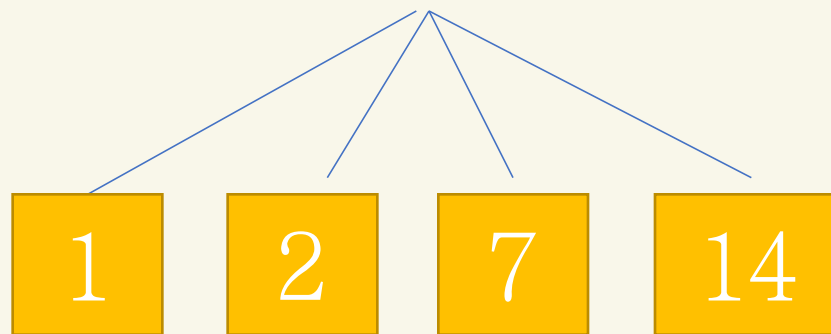
质数

13



不是质数

14



质数

下列哪些数是质数？

1 2 4 5 8 9 11

质数

如果一个数 n 是质数，它的约数是1和 n

如果一个数 n 是质数，那么 $2 \sim (n-1)$ 之间没有 n 的约数

质数

如何判断一个数是否是质数？

质数

判断一个数是否是质数

Step1: 假设

n 就是质数

Step2: 找约数

从2到 $(n-1)$ 找 n 的约数

标记找约数的结果

Step3: 判断标记结果

如果标记的结果找到了约数 n 不是质数

否则 n 就是质数

质数

判断一个数是否是质数

```
int n;  
cin >> n;  
bool flag = true;  
for (int i = 2; i <= n - 1; i++)  
{  
    if (n % i == 0)  
    {  
        flag = false;  
    }  
}  
if (flag == true)  
{  
    cout << "y";  
}  
else  
{  
    cout << "n";  
}
```

质数

程序优化

- 1：循环到根号 n 。(比如9，根号9为3，到3停止即可)
- 2：找到因数之后，结束循环
- 3：如果 n 大于1的情况下才是质数，否则不是

质数

```
int n;  
cin >> n;  
bool flag = true;  
for (int i = 2; i * i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        flag = false;  
        break;  
    }  
}  
if (flag == true && n > 1)  
{  
    cout << "y";  
}  
else  
{  
    cout << "n";  
}
```



Part 4

cmath模块

cmath模块

头文件： `#include <cmath>`

函数	返回值类型	函数功能	举例	结果
<code>pow(x,n)</code>	double	计算x的n次方	<code>pow(2,10)</code>	1024
<code>sqrt(x)</code>		对x取算术平方根	<code>sqrt(100)</code>	10
<code>ceil(x)</code>		取不小于等于x的整数	<code>ceil(6.3)</code>	7
<code>floor(x)</code>		取不大于等于x的整数	<code>floor(9.9)</code>	9
<code>round(x)</code>	和x保持一致	对x四舍五入到个位	<code>round(3.5)</code>	4
<code>abs(x)</code>		取x的绝对值	<code>abs(-123)</code>	123



Part 5

总结回顾

总结回顾

`break;`

在程序中表示**结束循环**

`continue;`

在程序中表示**跳过本次循环，继续下一次循环**

总结回顾

输出一个数所有的约数

求
n
所有
的
约
数

```
int n;  
cin >> n;  
for (int i = 1; i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        cout << i << " ";  
    }  
}
```

for
循环
和
if
嵌套

总结回顾

输出一个数的约数个数

求 n 所有的约数个数

```
int n, cnt = 0;  
cin >> n;  
for (int i = 1; i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        cnt++;  
    }  
}  
cout << cnt;
```

过程性：统计

结果性：输出

for 循环和 if 嵌套

总结回顾

判断一个数是否是质数

```
int n;  
cin >> n;  
bool flag = true;  
for (int i = 2; i * i <= n; i++)  
{  
    if (n % i == 0)  
    {  
        flag = false;  
        break;  
    }  
}  
if (flag == true && n > 1)  
{  
    cout << "y";  
}  
else  
{  
    cout << "n";  
}
```

总结回顾

头文件： `#include <cmath>`

函数	返回值类型	函数功能	举例	结果
<code>pow(x,n)</code>	double	计算x的n次方	<code>pow(2,10)</code>	1024
<code>sqrt(x)</code>		对x取算术平方根	<code>sqrt(100)</code>	10
<code>ceil(x)</code>		取不小于等于x的整数	<code>ceil(6.3)</code>	7
<code>floor(x)</code>		取不大于等于x的整数	<code>floor(9.9)</code>	9
<code>round(x)</code>	和x保持一致	对x四舍五入到个位	<code>round(3.5)</code>	4
<code>abs(x)</code>		取x的绝对值	<code>abs(-123)</code>	123

A horizontal yellow banner with blue triangular ends, centered on a solid blue background. The text "Class is over" is written in blue within the banner.

Class is over