

太戈编程
etiger.vip

信奥算法

变量作用域

局部变量

变量作用域

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int x=1;
4 int main(){
5     cout<<x<<endl;
6     int x=2;
7     cout<<x<<endl;
8     if(1){
9         int x=3;
10        cout<<x<<endl;
11    }
12    cout<<x<<endl;
13    return 0;
14 }
```

对第3,4,5行有作用

对第6,7,8,12,13行有作用

对第9,10行有作用

局部变量没有初始化

请预测
输出结果
老师运行

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int a,b,c,d,e,f;
4  int main(){
5      cout<<a<<" "<<b<<" ";
6      cout<<c<<" "<<d<<" ";
7      cout<<e<<" "<<f<<endl;
8      int u,v,w,x,y,z;
9      cout<<u<<" "<<v<<" ";
10     cout<<w<<" "<<x<<" ";
11     cout<<y<<" "<<z<<endl;
12     return 0;
13 }
```

全局变量
自动清零

局部变量
数值不确定

自定义函数

自己定义函数功能

有参数+有返回值

绝对值函数:整数

输入整数x，输出x的绝对值(absolute value)。 $-100 \leq x \leq 100$

输入样例:

-12

输出样例:

12

输入样例:

100

输出样例:

100

输入样例:

0

输出样例:

0

```
1 #include<iostream>
2 using namespace std;
3
4
5
6
7 int main(){
8     int x;
9     cin>>x;
10    cout<<abs(x);
11    return 0;
12 }
```

自定义abs
函数功能

使用
abs功能

绝对值函数:整数

函数
返回值
类型

函数
名称

参数
列表

```
3 int abs(int y){  
4     if(y>0) return y;  
5     else return -y;  
6 }
```

定义abs函数:
参数y为一个整数
返回一个整数答案
代表y的绝对值

若y大于0, 将y返回
否则,将-y作为答案返回

return操作将一个
答案值返回到调用
abs函数的位置

函数执行顺序

```
→ 1 #include<iostream>
→ 2 using namespace std;
→ 3 int abs(int y){
→ 4     if(y>0) return y;
→ 5     else return -y;
→ 6 }
→ 7 int main(){
→ 8     int x;
→ 9     cin>>x;
→10     cout<<abs(x);
→11     return 0;
→12 }
```

1,2,7,8,9,10

3,4,5,6

10,11,12

自定义函数

自己定义函数功能

无参数+无返回值

顏文字 *kaomoji*

打开"颜文字"程序

运行并观察结果

理解程序每一行

无返回值
void

```
1  #include<iostream>
2  using namespace std;
3  void cat(){
4      cout<<"m=^o^=m"<<endl;
5      cout<<"(=^ ^=)"<<endl;
6  }
7  void happy(){
8      cout<<"q(*^____^*)p"<<endl;
9      cout<<"=={*0o0*}=="<<endl;
10 }
11 int main(){
12     happy();
13     cat();
14     happy();
15     return 0;
16 }
```

自定义新
函数功能

void表示
无返回值

圆括号()
内无参数

使用
新功能

跪在地上哭

打开"跪在地上哭"

运行并观察结果

理解程序每一行

无返回值
void

```
1  #include<iostream>
2  using namespace std;
3  void cry(int n){
4      for(int i=0;i<n;i++)
5          cout<<"5";
6  }
7  void orz(int n){
8      for(int i=0;i<n;i++)
9          cout<<" 0|--|_ ";
10 }
11 int main(){
12     cry(100);
13     orz(200);
14     cry(100);
15     return 0;
16 }
```

自定义新
函数功能

void表示
无返回值

参数n
代表次数

使用
新功能

我没哭



4类函数定义

无返回值
无参数

```
void happy()  
void cat()
```

无返回值
有参数

```
void orz(int n)  
void cry(int n)
```

有返回值
无参数

```
int main()
```

有返回值
有参数

```
int max(int x, int y)  
double fabs(double x)
```

自定义

阶乘函数

```
long long f(int m)
```

计算m的阶乘

阶乘函数

输入非负整数 n ，输出 n 的阶乘(factorial)，数据范围： $0 \leq n \leq 20$

$$n \text{ 的阶乘} \quad n! = n \times (n - 1) \times \cdots \times 2 \times 1$$

输入样例：

5

输出样例：

120

输入样例：

10

输出样例：

3628800

输入样例：

20

输出样例：

2432902008176640000

记
笔
记

long long范围内
最大可以计算20阶乘

阶乘函数

补全
程序

```
1  #include<iostream>
2  using namespace std;
3  [ ] f(int m){
4      long long ans=1;
5      for(int i=2;i<=m;i++)
6          [ ]
7      return ans;
8  }
9  [ ] int main(){
10     int n;
11     cin>>n;
12     cout<<f(n)<<endl;
13     return 0;
14 }
```

f(n)代表n的阶乘

自定义

闰年函数

```
bool isLeap(int y)
```

判断y是否为闰年

判断闰年

口诀：四年一闰，百年不闰，四百年又闰

2020是闰年

2019不是闰年

2018不是闰年

2017不是闰年

2016是闰年

2000是闰年

1900不是闰年

1800不是闰年

1700不是闰年

1600是闰年

闰年函数

输入n个4位年份，判断每一年是不是闰年
用Yes或No表示结果。n≤100

输入样例

5

2020

2021

2024

1900

2000

输出样例

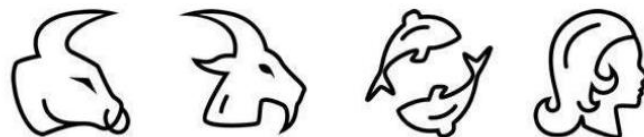
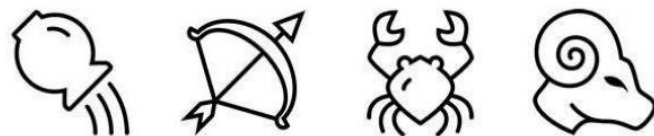
Yes

No

Yes

No

Yes



闰年函数

isLeap(y)
判断y是否为闰年

```
3 bool isLeap(int y){  
4     return y%400==0 || y%4==0&& y%100!=0;  
5 }
```

函数返回值判断以下条件是否成立：
y是400的倍数，
或者y是4的倍数且y不是100的倍数

闰年函数

```
3 bool isLeap(int y){
```

```
8 int main(){
```

```
9     int n,y;
```

```
10    cin>>n;
```

```
11    for(int i=0;i<n;++i){
```

```
12        cin>>y;
```

```
13        if(isLeap(y))
```

```
14            cout<<"Yes"<<endl;
```

```
15        else
```

```
16            cout<<"No"<<endl;
```

```
17    }
```

```
18    return 0;
```

```
19 }
```

闰年函数

为了测试函数正确性
可以用不同参数调用函数

```
1  #include<iostream>
2  using namespace std;
3  bool isLeap(int y){
8  int main(){
9      cout<<isLeap(1900)<<endl;
10     cout<<isLeap(2000)<<endl;
11     cout<<isLeap(2020)<<endl;
12     cout<<isLeap(2021)<<endl;
13     cout<<isLeap(2022)<<endl;
14     cout<<isLeap(2023)<<endl;
15     cout<<isLeap(2024)<<endl;
16     return 0;
17 }
```

自定义

回文函数

bool isHW(string s)

判断s是否为回文字符串

回文函数

输入n个正整数，判断每个数是否为回文数。
结果为Yes或No。n≤100，数字均小于 10^{18}

输入样例

5

2020

12321

567765

1991

2000000

输出样例

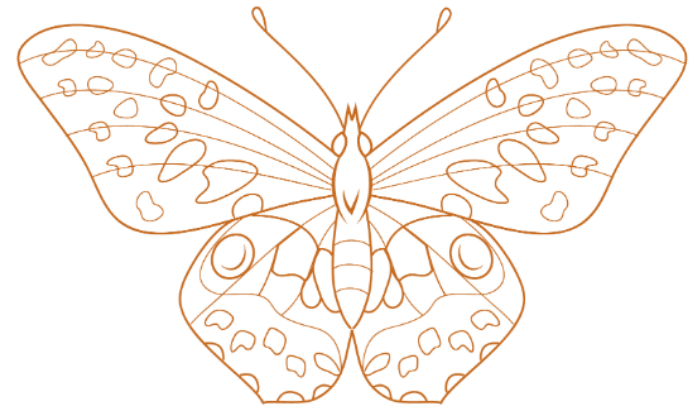
No

Yes

Yes

Yes

No



输入数字用
什么类型储存?

回文函数

```
14 int main(){
15     int n;
16     cin>>n;
17     for(int i=0;i<n;++i){
18         string s;
19         cin>>s;
20         if(isHW(s))
21             cout<<"Yes"<<endl;
22         else
23             cout<<"No"<<endl;
24     }
25     return 0;
26 }
```

虽然题目描述为数字
但是用字符串更方便

回文函数

```
3 bool isHW(string s){  
4     int l=0;  
5     int r=s.size()-1;  
6     while(l<r){  
7         if(s[l]!=s[r])  
8             return 0;  
9         l++;  
10        r--;  
11    }  
12    return 1;  
13 }
```

isHW(s)判断
s是否为回文字符串

左游标l

右游标r

当左右游标没相遇就循环重复

若左右字符不同

返回：不是回文

左游标右移1位

右游标左移1位

返回：是回文

回文函数

为了测试函数正确性
可以用不同参数调用函数

```
1  #include<iostream>
2  using namespace std;
3  bool isHW(string s){
14 int main(){
15     cout<<isHW("8")<<endl;
16     cout<<isHW("66")<<endl;
17     cout<<isHW("121")<<endl;
18     cout<<isHW("233")<<endl;
19     cout<<isHW("1232")<<endl;
20     cout<<isHW("12321")<<endl;
21     cout<<isHW("123421")<<endl;
22     return 0;
23 }
```

自定义

质数函数

```
bool isPrime(int x)
```

判断x是否为质数

质数

一个大于1的自然数 n ，如果只能被 n 自己和1整除，而不能被其他数整除，那么 n 就是个质数。

2是质数
3是质数
5是质数
7是质数
11是质数
13是质数
17是质数
19是质数
23是质数

$$4=2*2$$

$$6=2*3$$

$$8=2*4, 9=3*3, 10=2*5$$

$$12=2*6$$

$$14=2*7, 15=3*5, 16=2*8$$

$$18=2*9$$

$$20=2*10, 21=3*7, 22=2*11$$

$$24=2*12, 25=5*5, 26=2*13, 27=3*9, 28=2*14$$

4,6,8,9,10,12,14,15,16,18,20,21,22,24,
25,26,27,28 都不是质数, 是合数

质数判定函数1

```
1  #include<iostream>
2  using namespace std;
3  bool isPrime(int x) {
4      if(x<=1)return 0;
5      for(int i=2;i<=x-1;i++)
6          if(x%i==0)return 0;
7      return 1;
8  }
9  int main() {
10     int n;
11     cin>>n;
12     if(isPrime(n)) cout<<"Yes"<<endl;
13     else cout<<"No"<<endl;
14     return 0;
15 }
```

自定义函数
isPrime

返回值代表x
是否为质数

若x小于等于1,一定不是质数

循环查看可能的约数i

若x是i的倍数,确定x不是质数

若x没有任何约数,那么x是质数

质数判定函数2

```
1  #include<iostream>
2  using namespace std;
3  bool isPrime(int x) {
4      if(x<=1)return 0;
5      for(int i=2;i*i<=x;i++)
6          if(x%i==0)return 0;
7      return 1;
8  }
9  int main() {
10     int n;
11     cin>>n;
12     if(isPrime(n)) cout<<"Yes"<<endl;
13     else cout<<"No"<<endl;
14     return 0;
15 }
```

若n等于2数乘积
总有1数不超过 \sqrt{n}

假设n不是质数,分解
成两数乘积 $n=i*(n/i)$

取i为较小的约数
 $i \leq n/i$ 即 $i*i \leq n$

质数函数

为了测试函数正确性
可以用不同参数调用函数

```
1  #include<iostream>
2  using namespace std;
3  bool isPrime(int x) {
9  int main() {
10     cout<<isPrime(1)<<endl;
11     cout<<isPrime(2)<<endl;
12     cout<<isPrime(3)<<endl;
13     cout<<isPrime(4)<<endl;
14     cout<<isPrime(5)<<endl;
15     cout<<isPrime(12)<<endl;
16     cout<<isPrime(89)<<endl;
17     cout<<isPrime(91)<<endl;
18     return 0;
19 }
```

现场挑战

228

自定义函数
bool isPrime(int x)

返回值代表
x是否为质数

```
int main() {  
    int a,b;  
    cin>>a>>b;  
    if( a和b的差的绝对值为2并且a为质数并且b为质数 )  
        cout<<"Yes"<<endl;  
    else  
        cout<<"No"<<endl;  
    return 0;  
}
```

作业要求

- 0.翻译每一行写注释，解释函数含义，变量含义
- 1.每次出现错误都必须在微信班级群发错误笔记，含姓名，题号,标题,错误原因,如何避免
- 2.将该笔记记录在自己电脑的文档里，作为经验累积

举例：

姓名：黄晓明

题号：12

标题：长方形体积

错误原因：类型选择错误，应该用double浮点数

避免方法：每次定义时思考数据类型，提交前测试多组自编输入数据检验

太戈编程

1780

1781

288

拓展题

1371, 119, 345