计算机学院 高级语言程序设计 课程实验报告

实验题目：模板特化、模板元编程		学号：202200130048
日期：2023.5.9	班级： 6	姓名： 陈静雯
Email：1205037094@qq.com		

实验步骤与内容：

1. 请利用模板元编程设计一个类模板 template<unsigned M, unsigned N> Permutation，内含一个枚举值 VALUE，Permutation<M, N>::VALUE 的值为排列数 PMN(从 N 个值中选 M 个值有多少种可能性)。请参考排列数的定义以及在第九章课件第 98 页给出的通过模板元编程实现阶乘来实现上述类模板。
2. 结合 PPT 例 10.2 做以下实验
3. PPT 例 10.7 给出了一个实验利用栈反向输出单词，请修改这个程序，实现以下功能。
4. PPT 例 10.8 通过优先级队列模拟了细胞分裂，实践一下。请问这个程序能不能顺利编译运行？不能的话看一下问题出在哪里，解决问题并回答为什么。
5. 例 10.16 介绍了如何通过 sort 函数直接对 vector 类中的元素进行排序，请将 sort 函数扩展到 Point 类中。

结论分析与体会：

1.

```
#include<iostream>
using namespace std;

template <unsigned N>  //N 相当于<可变参数>
struct Factorial { //计算 N 的阶乘, 用 struct 编写类模板
  enum {
    VALUE = N * Factorial<N - 1>::VALUE
  };
};

template <>
struct Factorial<0> { //设定 N = 0 时 N 的阶乘
    enum { VALUE = 1 };
};

template<unsigned M, unsigned N>
struct Permutation { //计算 N 的阶乘, 用 struct 编写类模板
  enum {
    VALUE = Factorial<N>::VALUE / Factorial<N - M>::VALUE
  };
};
```

```
int main(){
    cout<<Permutation<3, 25>::VALUE<<endl;
}
```

2.

（1）

```
xe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
6
36      8
64      9.9
98.01
PS D:\code repository\code>
```

（2）

```
int main() {
    double a[]={1,2,3,4,5,6};
    double b[6]={0};
    transform(a,a+6,b,square);
    for(int i=0;i<6;i++)cout<<b[i]<<" ";
    cout << endl;
    return 0;
}
```

```
xe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
1 4 9 16 25 36
PS D:\code repository\code>
```

（3）

```
int main() {
    vector<double>a={1,2,3,4,5,6};
    vector<double>b;
    b.resize(a.size());
    transform(a.begin(),    a.end(),
       b.begin(),square);
    for(int i=0;i<6;i++)cout<<b[i]<<" ";
    cout << endl;
    return 0;
}
```

```
xe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
1 4 9 16 25 36
PS D:\code repository\code>
```

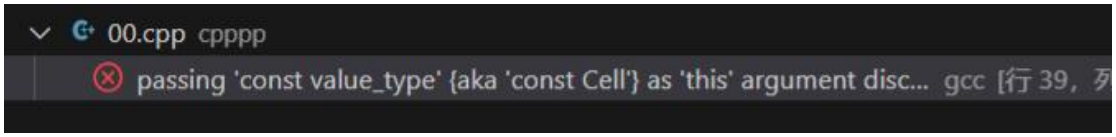
3.

```
int main() {
    stack<char> s;
     stack<char> s1;
    string str;
    cin >> str;   //从键盘输入一个字符串
    //将字符串的每个元素顺序压入栈中
    for (string::iterator iter = str.begin(); iter != str.end(); ++iter)
        s.push(*iter);
    //将栈中的元素顺序弹出并输出
     str.clear();
    while (!s.empty()) {
         char c = s.top();
         while(!s.empty()&&c!=','){
             s1.push(c);
             s.pop();
             if(!s.empty()) c = s.top();
         }
         if(!s.empty()) s.pop();
         while(!s1.empty()){
             str+=s1.top();
             s1.pop();
         }
         cout<<str<<endl;
         str.clear();

    }
    cout << endl;
    return 0;
}
```

```
xe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
terry,is,good
good
is
terry
```

4.



```
xe=D:\mingw64\bin\gdb.exe' '--interpreter=mi'
Simulation time: 5000
1296s: Cell #0 splits to #1 and #2
1840s: Cell #2 splits to #3 and #4
3042s: Cell #3 splits to #5 and #6
3240s: Cell #1 splits to #7 and #8
3373s: Cell #4 splits to #9 and #10
4556s: Cell #8 splits to #11 and #12
4663s: Cell #6 splits to #13 and #14
4690s: Cell #9 splits to #15 and #16
4869s: Cell #10 splits to #17 and #18
4914s: Cell #5 splits to #19 and #20
```

修改

```
    cellQueue.push(cell(0));     //将第一个细胞压入优先
    while (cellQueue.top().getSplitTime() <= t) {
        Cell c(cellQueue.top());
        c.split();  //模拟下一个细胞的分裂
        cellQueue.pop();                    //将刚刚分裂的细胞
```

原因：top（）返回的是 this 指针，this*指向的 cell 是 const 修饰的，是常量，不能调用非常量函数。

5.

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;
class Point{
public:
    int x;
    int y;
    Point(int a,int b){
        x=a;
        y=b;
    }
    void show(){
        cout<<x<<" "<<y<<endl;
    }
    ~Point(){}
};
bool cmd(Point a,Point b){
    if(a.x>b.x) return false;
    else if(a.x==b.x){
        if(a.y>=b.y){
            return false;
        }
```

```
        else{
            return true;
        }
    }
    else{
        return true;
    }
}
int main(){
    vector<Point>  pts  =  {Point(1,2),  Point(4,10),  Point(5,1),
Point(1,10), Point(3,2),Point(2,6), Point(4,3),Point(2,1)};
    sort(pts.begin(),pts.end(),cmd);
    for(vector<Point>::iterator i=pts.begin();i!=pts.end();i++){
        i->show();
    }
}
```