

## 运用向量求10000!的阶乘

我们知道，电脑里面的10000的数阶乘结果肯定是不能用int类型存储的，也就是说，平常的方法是求不出这个结果的。下面，我介绍一些用向量来模拟这个算法，其中向量里面的每一位都是代表一个数。

```
#include <iostream>
#include <vector>

using namespace std;

//就是n的阶乘
void calculate(int n){
    vector<int> v;
    v.push_back(1);

    unsigned int height = 0;
    int i = 1;
    int j = 0;
    unsigned int temp = 0;

    for(; i <= n; i++){
        height = 0;
        for(j = v.size() - 1; j >= 0; j--){
            temp = v[j] * i + height;
            height = 0;
            if(temp > 9){//说明进位了,取得高位,也就是进位
                height = temp / 10;
            }
            v[j] = temp % 10;
        }
        if(j < 0){
            if(height != 0){
                while(height){
                    v.insert(v.begin(), height % 10);
                    height /= 10;
                }
            }
        }
    }

    for(j = 0; j < v.size(); j++){
        cout << v[j];
    }
}
```

```
}  
cout << endl;  
}  
  
int main(){  
  
    calculate(10000);  
    return 0;  
}
```

别说10000，好几十万都能求，不过貌似时间比较慢，不知道大家觉得里面有没有优化的地方？  
结果：10000! = 501022688901891016735720586614100117236647626578353963642978190  
11647056170279631.....  
由于太长，这里省了。

本博客文章除特别声明，全部都是原创！  
转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)  
本文链接: [【】 \( \)](#)