```cpp
#include <iostream>
using namespace std;
//高精度的加  减  乘  第一步逆置转整
string minus_s(string a,string b){//字符串数字相减
    //判断大小关系 先考位数 再看字典序
    string c="";
    int lena=a.size(),lenb=b.size();
    if(lena<lenb) {string t=a;a=b;b=t;c+="-";swap(lena,lenb);}
    else if(lena==lenb){
        if(a<b) {string t=a;a=b;b=t;c+="-";}
        else if(a==b){
            return "0";
        }
    }
    //永远做a-b
    int x[3000]={0},y[3000]={0},z[3000]={0};//存z减法答案
    //字符串数字a 转化位整数数位同时倒过来 到x中
    for(int i=lena-1,j=0;i>=0;i--,j++)  x[j]=a[i]-'0';
    for(int i=lenb-1,j=0;i>=0;i--,j++)  y[j]=b[i]-'0';
    //x>y
    int len=max(lena,lenb),t=0;//当前借位量
    for(int i=0;i<=len;i++){
        z[i]=x[i]-y[i]-t;
        if(z[i]<0){
            z[i]+=10;
            t=1;
        }else{
            t=0;
        }
    }
    int flag=0;
    for(int i=len;i>=0;i--){
        if(z[i]!=0) flag=1;
        if(flag==1) c+=(z[i]+'0');
    }
    return c;
}

string plus_s(string a,string b){//大数加法
    int x[3000]={0};
    int y[3000]={0};
    int z[3000]={0};
    int lena=a.size(),lenb=b.size();
    for(int i=lena-1,j=0;i>=0;i--,j++){
        x[j]=a[i]-'0';
    }
    for(int i=lenb-1,j=0;i>=0;i--,j++){
        y[j]=b[i]-'0';
    }
    int t=0;
    int len=max(lena,lenb);
    for(int i=0;i<=len;i++){
```

```
        z[i]=x[i]+y[i]+t;
        if(z[i]>=10){
            z[i]-=10;
            t=1;
        }else{
            t=0;
        }
    }
    string c="";
    int k;
    for(int i=len;i>=0;i--){
        if(z[i]!=0){
            k=i;
            break;
        }
    }
    for(int i=k;i>=0;i--)c+=(z[i]+'0');
    return c;
}

string Mul_s(string a,string b){//大数乘法
    string c="";
    int lena=a.size(),lenb=b.size();
    int x[3000]={0},y[3000]={0},z[3000]={0};
    for(int i=lena-1,j=0;i>=0;i--,j++)  x[j]=a[i]-'0';
    for(int i=lenb-1,j=0;i>=0;i--,j++)  y[j]=b[i]-'0';
    for(int i=0;i<lena;i++){//x数组
        for(int j=0;j<lenb;j++){//y数组
            z[i+j]+=x[i]*y[j];
            if(z[i+j]>=10){
                z[i+j+1]+=z[i+j]/10;
                z[i+j]%=10;
            }
        }
    }
    int flag=0;
    for(int i=lena+lenb;i>=0;i--){
        if(z[i]!=0) flag=1;//从高往低遇到第一个不为0的数字
        if(flag==1) c+=z[i]+'0';
    }
    return c;
}

string Div_s(string a,long long b){//高精度除单精度
    int lena=a.size();
    int x[3000]={0},z[3000]={0};
    for(int i=lena-1;i>=0;i--)  x[i]=a[i]-'0';
    long long t=0;//用t去除
    for(int i=0;i<lena;i++){
        t=t*10+x[i];
        z[i]=t/b;
        t=t%b;
    }
    string c="";
    int flag=0;
```

```
    for(int i=0;i<lena;i++){
        if(z[i]!=0) flag=1;//从高往低遇到第一个不为0的数字
        if(flag==1) c+=z[i]+'0';
    }
    return c;
}

int main(){
    string s="1";
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        int m=i;
        string t="";
        while(m!=0){
            char tmp=(m%10+'0');
            t=tmp+t;
            m/=10;
        }
        s=Mul_s(s,t);
    }
    cout<<s;

}
```