

高精度加法

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=520;
int a[maxn],b[maxn],c[maxn];
int main(){
    string A,B;
    cin>>A>>B;
    int len=max(A.length(),B.length());
    int j=1;
    for(int i=A.length()-1;i>=0;i--){
        a[j++]=A[i]-'0';
    }
    j=1;
    for(int i=B.length()-1;i>=0;i--){
        b[j++]=B[i]-'0';
    }
    for(int i=1;i<=len;i++){
        c[i]=a[i]+b[i];
        c[i+1]+=c[i]/10;
        c[i]=c[i]%10;
    }
    if(c[len+1]!=0)
        len++;
    for(int i=len;i>=1;i--){
        printf("%d",c[i]);
    }
}
```

减法

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=100005;
int a[maxn],b[maxn],c[maxn],flag=1;
int main(){
    string A,B;
    cin>>A>>B;
    int la=A.length(),lb=B.length();
    if(la<lb||(la==lb&&A[0]<B[0])){
        swap(A,B);
        swap(la,lb);
        cout<<"-";
    }
    int j=1;
    for(int i=la-1;i>=0;i--){
        a[j++]=A[i]-'0';
    }
    j=1;
    for(int i=lb-1;i>=0;i--){
        b[j++]=B[i]-'0';
    }
    for(int i=1;i<=la;i++){
        if(a[i]<b[i]){
            a[i]+=10;
            a[i+1]--;
        }
        c[i]=a[i]-b[i];
    }
}
```

```

    }
    while(c[la]==0&&la>1){
        la--;
    }
    for(int i=la;i>=1;i--){
        printf("%d",c[i]);
    }
}

```

乘法

```

#include <bits/stdc++.h>
using namespace std;
const int maxn=5010;
int a[maxn],b[maxn],c[maxn];
int main(){
    string A,B;
    cin>>A>>B;
    int lena=A.length(),lenb=B.length();
    int j=1;
    for(int i=A.length()-1;i>=0;i--){
        a[j++]=A[i]-'0';
    }
    j=1;
    for(int i=B.length()-1;i>=0;i--){
        b[j++]=B[i]-'0';
    }
    for(int i=1;i<=lena;i++){
        for(int j=1;j<=lenb;j++){
            c[i+j-1]=c[i+j-1]+a[i]*b[j];
        }
    }
    int len=lena+lenb;
    for(int i=1;i<=len;i++){
        c[i+1]=c[i+1]+c[i]/10;
        c[i]=c[i]%10;
    }
    while(c[len]==0){
        len--;
    }
    for(int i=max(len,1);i>=1;i--){
        cout<<c[i];
    }
}

```