**String Multiplication**

class Solution {

public:

string multiply(string num1, string num2) {

if(num1=="0" || num2=="0") return "0";

int num1len = num1.length();

int num2len = num2.length();

vector<int>result(num1len+num2len, 0);

for(int i=num1len-1; i>=0; i--){

for(int j=num2len-1; j>=0; j--){

int product = (num1[i]-'0')\*(num2[j]-'0');

int sum = product + result[i+j+1];

result[i+j+1] = sum%10;

result[i+j] += sum/10;

}

}

string ans = "";

for(auto it:result){

if(!(ans.empty() && it==0)){

ans.push\_back(it+'0');

}

}

return ans.empty() ? "0" : ans;

}

};

**Other way**

class Solution {

public:

    string multiply(string num1, string num2) {

        string ProductResult="";

    bool Zeroes = true;

    int n=num1.length();

    int m=num2.length();

    vector<int> ProductStored(n+m,0);

        for(int it1=n-1;it1>=0;--it1){

            int pickedDigit1 = num1[it1]-'0';

            for(int it2=m-1;it2>=0;--it2){

                int pickedDigit2 = num2[it2]-'0';

                int multiplication = ProductStored[it1+it2+1]+pickedDigit1\*pickedDigit2;

                ProductStored[it1+it2+1]=multiplication%10;

                ProductStored[it1+it2] +=multiplication/10; // handle carries

            }

        }

        for(int Product : ProductStored){

            if(Product==0 && Zeroes ) continue; // skip leading Zeros

            Zeroes=false;

            ProductResult= ProductResult+ to\_string(Product);

        }

        if (ProductResult.empty()) ProductResult = "0";

        return ProductResult;

    }

};