**Mulitplication of large integers**

#include <iostream>

#include <string>

#define OVERFLOW 2

#define ROW b\_len

#define COL a\_len+b\_len+OVERFLOW

using namespace std;

int carry(int num) {

int carry = 0;

if(num>=10) {

while(num!=0) {

carry = num %10;

num = num/10;

}

}

else carry = 0;

return carry;

}

int num(char a) {

return int(a)-48;

}

string mul(string a, string b) {

string r;

int a\_len = a.length();

int b\_len = b.length();

int mat[ROW][COL];

for(int i =0; i<ROW; ++i) {

for(int j=0; j<COL; ++j) {

mat[i][j] = 0;

}

}

int c=0, n,x=a\_len-1,y=b\_len-1;

for(int i=0; i<ROW; ++i) {

x=a\_len-1;

c = 0;

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

if((x>=0)&&(y>=0)) {

n = (num(a[x])\*num(b[y]))+c;

mat[i][j] = n%10;

c = carry(n);

}

else if((x>=-1)&&(y>=-1)) mat[i][j] = c;

x=x-1;

}

y=y-1;

}

c = 0;

int sum\_arr[COL];

for(int i =0; i<COL; ++i) sum\_arr[i] = 0;

for(int i=0; i<ROW; ++i) {

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

sum\_arr[j] += (mat[i][j]);

}

}

int temp;

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

sum\_arr[i] += c;

temp = sum\_arr[i];

sum\_arr[i] = sum\_arr[i]%10;

c = carry(temp);

}

for(int i=0; i<COL; ++i) {

r.push\_back(char(sum\_arr[i]+48));

}

while(r[0]=='0'){

r = r.substr(1,r.length()-1);

}

return r;

}

void printhuge(string a) {

cout<<"\n";

for(string::iterator i = a.begin(); i!=a.end(); ++i) {

cout<<\*i;

}

}

int main() {

string a,b;

cin>>a>>b;

printhuge(mul(a,b));

return 0;

}