# Problem 1A:

* Check whether there is a possibility of output going out of range , and insert long long int where ever it is possible to do so.

# Problem 1B:

* There will be some special numbers on the shoulders of which the whole program runs , do not miss the case where the input may be some how related to that number for example, the input may be multiple of that **special number**
* In some questions there may be some variables such as remainder , etc ,do not miss the input case for which such variables take **zero value**

# Problem 1C:

* Do write some smart macros to avoid cumbersome codes Ex:

#define D double

#define S(x) ((x)\*(x))

#define G(t) a[t]=x[t]-x[2];b[t]=y[t]-y[2];c[t]=(S(x[t])+S(y[t])- S(x[2])-S(y[2]))/2;

#define M(p,q) (p[0]\*q[1]-p[1]\*q[0])

* **GCD** for **doubles** : D g(D a,D b){return fabs(b)<1e-4?a:g(b,fmod(a,b));}

More the precision i.e 1e-x ;x less accurate the answer will be.

# Problem 2A:

Map can be created as follows map<string,int> p; and input of strings and integers can be taken by declaring integer arrays and string arrays;

If you want to find out the index at which maximum of an array of values occur,then for loop is as follows

for(i=0;p[s[i]]<m;i++);

if you want to end the for loop at maxima and at which some other condition C

then for loop is as follows

for(i=0;p[s[i]]<m||C;i++);

* NOTE:#include<bits/stdc++.h>-contains every library of c++

Contests:

Pop() always gives segmentation fault when the container is empty.

Deque is always better at inserting than vector.

When insertions are more if vector is used then we get time limit exceeded, when [] operators are more if we used deque istead of vector we get runtime error.

**16th may 2016:**

1) Its pretty much disturbing that there is a function to\_string(int) in strings library . 123456789101112131415….

This was the problem to find nth digit. Creating a string and using to\_string was the smartest thing to do.

2) nan error(not quite a number) occurs for long double ex: when I changed long double to double I got the right answer.

3) multiset is an pain in the ass container cuz the begin(),etc. iterators points to rvalue(I mean not lvalue)so we cant change the values inserted in multiset once we’ve inserted.

**17th may 2016:**

1. Sort can be used for arrays also Ex: int a[n]; sort(a,a+n)
2. Worst problem ever :haven’t even understood the algorithm

<http://codeforces.com/contest/671/problem/B>

3)\_\_gcd() in c++.

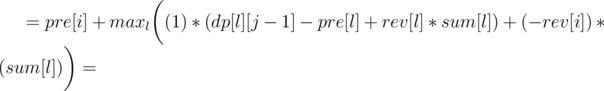
4) in questions where intervals of arrays are involved ,when we are obligated to check almost every element in an order, then don’t do it as if u are dumb, use dynamic programming and create the result from previous results , that decreased the time of execution from 2 s to 233 ms.

**18th may 2016:**

* 1) dividing n levels into k regions of levels by k-1 barriers. Try to use dp.
* Let *dp*[*i*][*j*] denote the optimal result for prefix of *i* levels, if we divide them into *j* regions.
* Let *pre*[*i*] denote the result for region containing levels 1, 2, ..., *i* (think how to calculate it easily with one loop).
* Let *sum*[*i*] denote the sum of *tj* for all 1 ≤ *j* ≤ *i*.
* Let *rev*[*i*] denote the sum of http://codeforces.com/predownloaded/ba/4d/ba4dc1db2b42181258933d9d543fd067b11187f2.png for all 1 ≤ *j* ≤ *i*.

Now let's write formula for *dp*[*i*][*j*], as the maximum over *l* denoting the end of the previous region:

http://codeforces.com/predownloaded/7d/db/7ddbb4d624eeb859ad374d59428f2d1ddc3f2090.png



http://codeforces.com/predownloaded/d8/72/d872bbf25d0b0c9635d4142f830015279ac17b70.png

675d:

You should see this one , money transfers, examples of usage of circular array.

**24th may 2016:**

Always use atoi , never use stoi FYI : arguments of stoi are string, where as args of atoi are char\*.pow function is a bitch never use that incase u want a double in return ,always define ur own power function.

1st June 2016:

Today I came across a wonderful and shortest implementation ever in segment trees tutorials.

Note: if p is even then p^1=p+1 ; p^1=p-1

[Segment trees](http://codeforces.com/blog/entry/18051)

2)learned about variadic functions and macros.

And found the best debugging tool which is a set of functions though.

3)The difference between for(auto it:s) and for(auto& it:s)

Is that in second one u can change the value of it in the loop.

**2nd June 2016:**

1. Sometimes if (! a\*b) doesn’t work.
2. I realized that two pointers method only works if the array’s size is around 10^5 and time limit Is 1 sec. If size is 10^9 the it needs 2 or 3 secs. Sometimes we have to generalize that 2 loops works for 10^9 only for 2 or 3 secs, works for 10^5 at a limit of 1 sec. the speciality of two pointers method is that even though it runs two nested loops it has O ( n ) complexity ,because the nested loop when run for the first time it takes time to catch up, when run for the remaining times it takes little lesser time because it uses the history of the previous run of loop to run this loop with proper boundaries.

20th july 2016:

1)I've noticed that returned value of a recursive function's function is to connect all the calls i.e it is the information passed from one call to another . So while writing recursive function only return the useful information. Source: topcoder data science tutorials ,intro to recursion part 1.

2)g++ -std=c++11 your\_file.cpp -o your\_program.....This is used to run g++ in c++ 11 mode.

3) One of the striking differences between deque and list is that deque contains operator[] but list doesnt.

4) The difference between erase and remove is that the argument of erase is the position or iterator , and that of remove is the value itself.

5)advance (it,distance\_n) function is used to advance the iterator. Similarly next and prev functions are in <iterator>