1200 - Horns and Hoofs

Search out the limit of the two goods.

For horn, you'll earn “Ax-x^2” when buys x units of it. It is very obvious that x=A/2 is the extremum point. Compare it with the limit of horn. The same to hoofs.

Notice that x must be an integer, so you could count x-1, x, x+1, and use the best among them.

1201 - Which day is it?

A boring Simulating & String Processing problem. WARNING: you should output [x] instead of [x].

1202 - Rectangles travel

Greedy Algorithm. First judge whether it’s impossible to go from one rectangle to another.

If there exists a way, Then count the y coordinate which is JUST able to go from one to another.

1203 - Conference

Comes from the old problem - Segment-Cover. Easy to find a Dynamic algorithm.

1204 - Idempotents

From the problem description, x(x-1)=0(mod pq).

If pq|x, 0 is the answer.

If pq|x-1, 1 is the answer.

If p|x&q|x-1, suppose x=Xp=Yq+1, then Xp-Yq=1. Count out X, then x is just here.

Similarly, if p|x-1&q|x, there’s another answer.

1205 - Underground

Regard every end point as vertex, then it’s Dijkstra problem.

1206 - Sum of digits

The answer is 36*55^(k-1), proving follows:

First you should see that S(A+B) = S(A) + S(B) just means, there's no digit carry in the addition between A and B.

If one number is 10, then the first digit of the other number could be 1~8, the second digit could be 0~9. 8*10 possibilities.

If one number is 11, then 8*9 possibilities.

…

If one number is 19, then 8*1 possibilities. 8*55 in total.

Similarly, 7*55 possibilities for 2x. 36*55 in total.

Coz the first digit could not be 0, so 36 possibilities for it, and 55 possibilities for any other digit.

1207 - Median

Take out the point A in the leftmost (if there're multi points, get the undermost one). Then sort all the other points by the polar angle to A. The median one is the answer.

1208 - Legendary teams contest

Data size is small, brute force in O(2^18) is okay. You could give it proper branch-cutting, then it takes no time.

1209 - 1, 10, 100, 1000...

Judge whether n is equal to i*(i-1)/2+1, here i is an integer.

Notice that n<2^31, so you'd better use "unsigned int", depending on your trivial processing.

1210 - Kind spirits

Obvious dynamic problem. From the bottom to the top, dp level after level.

1211 - Collective guarantee

No matter how complex it is in the description. It is very clear that your job is to judge whether there's accurate one "0", and no cycle. Use hash strategy with recursion, the complexity in time is O(N). Coz multi input, it's still not very fast.

1212 - Sea battle

My algorithm is like that, process both vertical & horizon (last ship's direction, if it is not 1x1). We talk about horizon as an example. For each line, scan the segment which was already covered by at most 30 ships. In at most O(30x30) time, you'll get the empty room for the last ship in this single line.

I'm sure that there's better algorithm, but no need for this problem.

1213 - Cockroaches!

Very difficult in the problem description. But our aim is to find a large enough tree in a connected graph! Just count the number of vertex N, then outputting N-1 is okay. Faint.

1214 - A strange procedure

```
y'=x*x+y

  x'=x*x+y'=x*x+x*x+y

  y' =sqrt(x'+(y'/abs(y'))*(-abs(y')))

     =sqrt(x'-y')

     =sqrt(x*x)

     =x

  x' = x'-2*y*y

     = x*x+x*x+y-x*x*2

     = y
```

It is equal to the swap of two values.

1215 - Exactness of projectile hit

First you should see that if the center of circle is inside the polygon, then you should output "0.000".

After that we could check every edge with the center of circle, there're three conditions:

It can be distinguished by judging the obtuse angle. Then find the smallest among them.

1216 - Two Pawns and One King

There're really a large number of special conditions in it. I don't suppose you to solve it. (Coz there's no algorithm element) But I remember the AC problem has been posted in the Ural Web board.

1217 - Unlucky Tickets

Consider a number with 8 digits: abcdefgh

Suppose a+b+c+d = e+f+g+h = a+c+e+g = b+d+f+h = i, then

b+d = e+g

a+c = f+h

Suppose b+d = e+g = j, then

a+c = f+h = i-j

Enumerate the possibilities of i and j, it's a combinatorics problem now. You could help yourself to extend n from 8 to any one.

1218 - Episode N-th: The Jedi Tournament

Construct a directed graph, then examine every vertex, whether it's okay to travel from it to any other vertex.

By definition, O(N^3) in time, but using adjacency list makes everything quicker.

1219 - Symbolic Sequence

My program just output random letters. It's obvious that you could AC mostly. I use srand(23405), and AC without any wrong answer:)

1220 - Stacks

In fact, it's an easy Stack-Simulating problem, but sadly, 4byte for data, 4byte for link, 800K>750K!

A way to avoid it depends on counting the pop times of every stack. I debugged on URAL during the contest for 2 hours, then I found out it's impossible to read the input file twice!

My way to save space before the rejudgement of URAL, is to compress pointer(4Byte) to 3Byte (coz it's in the range 1..1000000), then 700K in total. But after the rejudgement, compiler changes! (Maybe Pascal can successfully pass it, but I'm using C++)

At last, I come back to the original algorithm, using 400K+134K to record the input file. Then I process it from the last operation.

I use p[stack] to record the number of POP operations met (from the last operation), and prev[stack] denotes the position of the latest one. If I met a PUSH operation, then I'll know the data should be outputted for prev[stack] is just the data in this PUSH. Dec(p[stack]), and get the new prev[stack] (this could be got easily by re-use the 400K array, and record it everywhere).

1221 - Malevich Strikes Back!

Use right[i][j] to denote the number of grids which have the same color as (i,j).

Enumerate a grid (i,j) with black color, then use right[i][j]*2-1 to get the height of the possible diagonal square. And then in O(N) time, you could verify whether this square is really possible.

1222 - Chernobyl' Eagles

An old math problem, split 3 as many as possible. If 4 is left, don't split it into 1+3, remaining 2+2 or 4 is okay.

1223 - Chernobyl' Eagle on a Roof

A good dp problem. Use opt[i][j] to express the maximum number of floors that can be experimented with i times, j eggs.

It's easy to see that opt[i][j] = opt[i-1][j-1]+opt[i-1][j]. Compress $O(N^2)$ in space to O(N), then $O(N^2)$ in time is okay.

ZHU Chenguang in China has a thesis of it in CTSC2004, his complexity is about O(sqrt(N)) for large N.

1224 - Spiral

Not difficult to prove that if n<=m, the answer is 2*(n-1), or else, it's 2*(m-1)+1.

1225 - Flags

Use opt[i][0..1] to denote the number of answers for i in length, the last one is white or red.

Counting the recursion formula is yours, notice that you should tie the blue stripe along with the white or red one to its right.

1226 - esreveR redrO

The sample input file has been corrected, there's one more '-' in the past.

Just reverse every word :)

1227 - Rally Championship

Notice that the problem doesn't say that the starting(ending) point should be exactly on the node! So:

1. There's a cycle, output "YES".

2. There isn't any cycle, then it must be a tree or a forest. List every node as the starting point, and counting the deepest point makes the way.

1228 - Array

Just use module by definition in the problem description.

1229 - Strong Brickwork

In fact, there’s a very easy way to solve it. Coz n and m are both even, you could split it into 2x2 grids. In any one 2x2 grid, two bricks could be placed vertically or horizontally, and at least one is okay! (Can be proved by discussing THREE different conditions)

1230 - Introspective Program

The main idea is to use string($) to avoid the appearance of ‘ and “.

I’ll only say that ;)

1231 - Turing One, Two, Three, ...

This problem is VERY difficult to be understood. The so-called “condition” is just like LABEL in our programming language. I’ll describe it below:

If the current character is “Current character”, changes it to “New character”, perform “Movement”, and the condition changes to “New condition”.

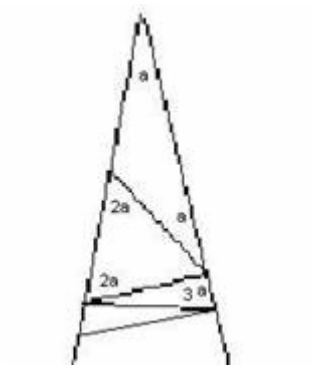
For current condition and character, look up the “control table”, and find the first line, which “Current condition” and “Current character” suffer the current value. Then this is the sentence to be processed. (If can’t find such sentence, the program exit.)

After that, it's an easy Josephus problem.

1232 - Asteroid Landing

For a fixed n (number of segment), like the picture to the right, we could count up the minimum height by the length of each segment - d, and alpha. And d*n is just the maximum
height. List all the possible n, if h is inside the interval, then this n is correct.

Now n is assured, we should get the coordinates. There must be a mathematical way, but I
don't know it. My solution is to use binary-search to guess the vertex angle, then check whether the height as the picture described is okay.



1233 - Amusing Numbers

An easy math problem. Such as 7435, it is in the position N.

First we should count the numbers in the range 1~9999, which is to the left of 7435, that's:

7+65+644+6436

If N is smaller than it, there won't be any solution. Or else, let N minus it.

Then let's count the numbers in the range 10000-99999, which is to the left of 7345, that's: 64350.

If N is smaller than it, we could use the exact place of it, to count the total. Or else, continue to see 100000~999999, …

1234 - Bricks

A boring math problem. Suppose A<=B<=C, D<=E.

If A<=D&&B<=E, YES.

If A>D&&B>E, NO.

Set dis to sqrt(sqr(A)+sqr(B)).

If A<=D, then let dis lie skew, just fills D, shorten the height as much as possible, and check it.

If B<=E, then let dis lie skew, just fills E, shorten the width as much as possible, and check it.

1235 - Cricket Field

The date size is not large, so we could just use enumeration.

Enumerate two point and get the x & y coordinate from them. Then check every point else, if it's to the upper right of (x,y), then update the maximum border length if it's larger.

1236 - Decoding Task

From the space (ASCII 32), and the password for it, we could get the 1st byte of the key, then the 1st byte of the text, etc..

1237 - Evacuation Plan

Only a better plan needed, so it's unnecessary to get the best scheme. In fact, only finding a cycle with weight lower than 0 is needed. Floyd is okay.

1238 - Folding

DP problem. Use F[i,j] to record the best folded sequence from ith to (i+j-1)th position (for the original string).

Pay attention to the space used, coz only 1000K provided, and F[i][j] is no longer than j, then we could list F[i][1..max] to a char array.

When we're considering the state transferring, there're three possibilities: 1. Original String; 2. SQ; 3. X(S).

1239 - Ghost Busters

A really hard analytic geometry problem.

Enumerate two spheres, determine the common tangent of them, which also passes the origin. Check all the lines like that. Just in case, I also check all the segment which goes through no only the origin by also the center of sphere.

1240 - Heroes Of Might And Magic

MinHP[i][j][k] denotes the minimum number of monsters (not in pack), when there're still i points of MP left, j points of HP left, and the monster is at position k.

From there, you could see the dynamic programming algorithm obviously.

Notice that, you should put the enemy as far as possible by listing all the possible positions!

of sphere

1241 - Inlay Cutters

Regard the borders as "cut"s, then it'll be much easier to be used.

Firstly, construct the graph, record whether there's "cut" in the 8 directions, and the number of "cut"s on this point.

Secondarily, you should pre-process. For each cross point (more than 2 "cut"s on it), compute the nearest cross point in 8 directions.

In the main program, enumerate a cross point, then check the four pairs of directions which could form a right angle. Then use the pre-process's info, it's very easy to check:)

1242 - Werewolf

This problem is much easier than the others in NEERC.

Find the victims, their ancestors and descendants are not werewolf.

1243 - Divorce of the seven dwarfs

Read the input file digit after digit, and do “mod” / “%” after reading every digit.

1244 - Gentlemen

01knapsack problem. Hash Dp will fit for it.

1245 - Pictures

An old problem. With one rectangle, the problem is easy.

With two, just enumerate a separated line, and count the smallest rectangle for each one of them.

This is $O(N^2)$ algorithm. You could get O(NlogN) algorithm by “interval tree”, but not necessary for this one.

1246 - A tethered dog

It’s very difficult to be understood. In fact, it just tells you to calculate whether the vertexes of a polygon is in clockwise order.

1247 - Check a sequence

Coz in the inequation's right

Ai + Ai-1 + ... + Aj <= (j – i + 1) + N

there's (j-i+1). We could let A'x = Ax - 1, then we're asked to judge if there exists

A'i + A'i-1 + ... + A'j > N

It is just a task to get the largest subsequence. Add continuously from A'1, if the current total is below 0, then make it as zero, and process on.

1248 - A sequence sum

The addition task for big real number. Notice you should round it before outputting.

1249 - Ancient Necropol

Map[i][j] denotes whether (i,j) is 1 or 0.

left[i][j]=0 for Map[i][j]=0

left[i][j] records the number of “1”s to its right for Map[i][j]=1

right[i][j]=0 for Map[i][j]=0

right[i][j] records the number of “1”s to its left for Map[i][j]=1

With the help of left and right array, I think the cute readers will get the answer.

Shteiner Sergei from Russia gave me an easy solution:

I think, that the solution is easier. You just should check every square 2*2 and if there is one, that contains exactly 3 '1' output ('NO') else output ('YES');

1250 - Sea Burial

A floodfill problem as you think, but maybe you haven't thought out a perfect way to color the girds. I'll help you:

Change '.' to 1, '#' to 2, borders to 0.

1. Floodfill from the point where the first shaman drowned, change the connected 1 to 3.

2. Floodfill from (0,0) - border - changing all the connected 1,2 or 0 to 3.
   (Now the positions which are not 3, forms the sacred parts).

3. Find all of the "2"s in this area, floodfill them to 3, the number of floodfill is just the number of islands.

1251 - Cemetery Manager

I contacted the author, and the problem description has been changed for three times. Please look at the current one, it's much more clear than before.

The main idea of the algorithm is to simulate. The important optimization I want to say it like that:

Use used[i] to record the number of used coffins in Line i.

Use min[i] to record the leftmost position of the empty grid in Line i. :) It's easy in the remaining mission.

1252 - Sorting the tombstones

Let's look at the sample

5

30

21

56

40

17

If we're asked to get a sequence in ascending order, then 17 should be at 1st place. But it's at 5th now. So K+1 should be a divisor of (5-1).

Similarly, count all of the numbers like (5-1) below, calculate the greatest common divisor of them, that is the upper limit of K+1.

QSort might make your program TLE, so please use HASH sort.

1253 - Necrologues

Recursion outputting is okay. Throw the endless loop and overlength program away.

1254 - Die Hard

Easy dijkstra might get TLE. Maybe you should use the standard dijkstra? With Fibonacci Heap!

May way to avoid it is that, with NO heap used, just use INSERT operator to insert the new born data to the correct position of an ordered array. Luckily, ACed:)

1255 - Graveyard of the Cosa Nostra

It’s easy to see that, one of the two best solutions is to empty a small rectangle (n mod k)*(n mod k)! Then the left parts will be filled.

Another way is like that:

---||

---||

|| ||

||---

||---

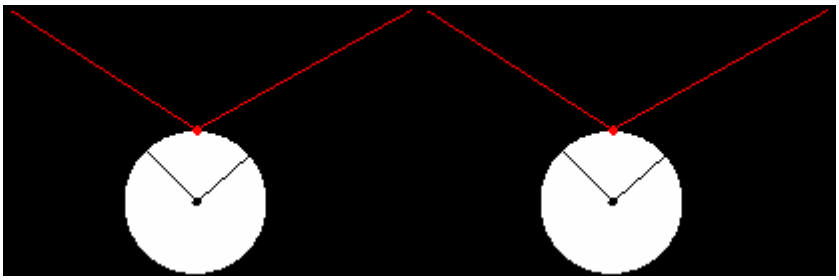
Now (n mod k)*(n div k)*4 rectangles placed. WARNING, you could still place more in the small square in the MIDDLE of it! That’s a smaller problem:)

1256 - Cemetery Guard

If the positions of two points and a circle are fixed, count the shortest path from one point to the circle, then to another point. Construct the function depending on the place of the turning point, I think that it’s a unimodal function, and trichotomous search could be used. (Even ACed, I don’t know the way to prove it, sorry)

Cosmin from Romania gave me an binary search algorithm and proving:

Consider the solution to be D, than the points on the elipse with sum=d and focal points A and B is tangent to the circle. So you can use binary search for determining D, and the function you need to see if D is smaller or bigger is check if the intersection of the elipse with the circle is void, or has two points or has exactly one point.



1257 - Hyphentation

A boring words processing problem, the only thick is like that:

pos('ab','cabab')=2, but if we meet 'cabab' in the end-of-line, maybe split it into 'cab'+'ab' is better than 'c'+'abab'.

1258 - Pool

It's not much difficult if you think everything deeply. Let's see the example:

One F, one B, two Ls and two Rs.

You must have thought out the algorithm depending on virtual image. In fact, (x1,y1) is moved W*4 to its left(*), and D*2 to its front(*). You'll find that soon:

4 and 2, must have something to do with the number of L/R and F/B respectively.

And the direction of (*) in the above paragraphs depends on the first appearance of L/R and F/B.

(The first appearance of F/B is F in the sample, then the direction of (*) is front, not back)

Warning if the total number of appearance of L/R or F/B is odd, it's not the simple shape like W*x, D*x, you should pay more attention on the coordinates' reflection.

1259 - How to become star

Let think the problem we are facing deeply, and find two examples:

When n=5:

1/5pi * 5 = pi

3/5pi * 5 = 3pi

They could form two stars. Here 1/5pi and 3/5pi represents the degree of α.

(The multiple of pis in the right side of equations must be odd, or else it won't form a star)

When n=9, there're four by definition.

1/9pi * 9 = pi

3/9pi * 9 = 3pi

5/9pi * 9 = 5pi

7/9pi * 9 = 7pi

But the second one is also tenable in 3/9 * 3 = pi, so it's already close when three edges are drawn.

It's very easy for odd n, the cute readers must have got the answer for it by the above paragraphs.

When n is even, firstly we can see the multiple of pis in the right side of equations must be even.

Notice when n=10:

2/10pi * 10 = 2pi => 2/10pi * 5 = 1pi

4/10pi * 10 = 4pi

6/10pi * 10 = 6pi => 6/10pi * 5 = 3pi

8/10pi * 10 = 8pi

And from 6/12pi *12 = 6pi => 6/12pi * 4 = 2pi we could see the conflict:

There're two conditions (regard the multiple of pis in the right side is x)

gcd(n/2,x/2)!=1 (for the second one);

gcd(n/2,x/2)==1, but gcd(n/2,2)==gcd(x/2)==1 (for the first one).

Aha, it's easy then, right?

1260 - A nudnik photographer

You can search some small test data, then the solution is here:

For ans[n], there're three ways to construct it.

Put 2 onto 2nd position, then it's ans[n-1];

Put 3 onto 2nd position, put 2 onto 3rd position. This is ans[n-3]! (You must put 4 onto 4th position)

The last condition contains only one possible way like that (first odd, then even)

13578642

So ans[n] = ans[n-1] + ans[n-3] + 1.

1261 - Tips

It's not easy to understand the description. I'll explain then.

For a fix amount of money, make it as the difference between two numbers. Such as 6 = 9-3.

And both of minuend and subtrahend should be sum of some 3^x-like number. And same x can't occur twice.

e.g. 3+27+243 is legal, but 3+3+27 and 3+4 are illegal.

And there's a special request, the subtrahend mustn't be zero!

It's a well-known math problem, first make the original number into trinary system.

Let it plus $(11111111111)_3$ (as long as possible). Now

the digit with 2 means that 3^x should occur in minuend;

the digit with 0 means that 3^x should occur in subtrahend;

the digit with 1 means that 3^x should not occur in both minuend and subtrahend.

Aha~ is that okay? No! If the subtrahend is zero? Think it over yourself :D

1262 - Pseudo-roman number

You must have known the definition of roman numerals. Let's see the ones place:

0	1	2	3	4	5	6	7	8	9
NIL	I	II	III	IV	V	VI	VII	VIII	IX

If this ones place is 8, we'll use 4 letters.

It's the same in other digits, just change the letter, without change of number of letters. Such as:

1 - I 10 - X

2 - I 20 - XX

1263 - Elections

Faint, count it yourself.

1264 - Workdays

I've nothing to say. Just output n*(m+1).

1265 - Mirror

Check whether two points are to the different sides of the mirror. Then calculate the virtual image, and connect them.

You'd better use vector product to avoid the precious error.

1266 – Kirchoff Laws

For Chinese students, you'd better see this book<**金牌之路解题指导>** p242 problem6.

In fact, there's a Kirchhoff theorem, it says that in every loop of the resistance circuit,

if we see electric current directed, then the sum of I*R should be ZERO in this loop.

The remaining course is to list equations.

First we use the apparent knowledge, for each connecting point, $I_{in} = I_{out}$. From that we can list exactly n-2 equations.

Regard connecting points as nodes, resistance as edge. Build a dfs tree, and add remaining edges.

For each added edge, we could get a loop. The total number of this kind of loops is exactly enough.

Gauss elimination is okay.

1267 - Yekaterinburg subway

Use $2^n * n$ to denote state. $0..2^n-1$ denotes the visited stations, 1..n denotes his current position.

opt[i][j]($0<=i<2^n$, $0<=j<n$) denotes the minimum time would be spent by him.

Let's see the phase, state transfer always makes i larger (opt[i][j]), so we can generate it from lower i to larger i.

And you should pay more attention on the starting station. There're two ways when he's on the first train:

going left or going right.

They might lead different starting time. So you must use dynamic programming twice!

1268 - Little Chu

I haven't thought out a good algorithm to solve this problem. I use "const" finally.

I have already changed the aim to that:

for a fixed prime p, calculate the minimum integer x, so p-1 is the minimum integer y, with $x^y = 1 (\mathrm{mod}\ p)$.

If you have got some good algorithms, write on our guestbook. Thank you very much.

a275417@movemail.com proved me this solution later 2004-8-19

If p is the number of days in a week, what you want is a generator of the group Zp*, i.e. such a number n, that $[n^{(p-1)/q}] \bmod p$ is not equal to 1 for all prime divisors q of (p-1). Try guessing and checking some values of n from the range 1...(p-1); the chances are that you will find a generator in at most log(p) guesses, so you have a $\log^2(p)$ algorithm.

1269 - Obscene words filter

For Chinese students, you can see my thesis in CTSC2004 training.

It has something to do with kmp, but we can use a tree instead of kmp's prefix array.

This is the finite automated machine. For every node, there should be a "failure pointer", just like kmp algorithm.

The largest difficulty is to minimal the space used. So you could allocate children pointer dynamically.

1270 - Unicube

Very easy in this contest. List the 24 possible rotation, and make edges between them by corresponding.

Matching is okay then. The coordinates are NO USE. Believe your understanding :D

1271 - Sailing directions

Regard one of three vertexes of each ship, as one of three vertexes of the Oil Boat.

Then there's n*9 points on the plane, some of them are illegal.

It's very easy to see, that the shortest path is only related to the points described above.

You'd better count the illegalness of such points, and the path between them VERY VERY carefully!!

A special notice, the starting and ending point might be illegal!

1272 - Non-Yekterinburg subway

You'll see the minimal spanning tree structure clearly. I don’t think I need to say anything more.

1273 - Tie

A little like the old problem in String Processing.

Use opt[i][j] to denote the maximum of ties can be placed without crossing from (0,0..i) to (1,0..j).

For each opt[i][j], consider opt[i-1][j], opt[i][j-1], and opt[i-1][j-1]+1 if (0,i) can be connected by (1,j).

1274 - Fractional Arithmetic

Boring problem. Read the fraction, process (add, plus, etc.), and reduce.

You'd better use Int64 to avoid crossing the limit of Longint.

There're many ways to be considered, 0, no fraction, no coefficient, etc.

Warning, don't forget <CR>

1275 - Knights of the Round Table

Our team failed to understand the problem well at the beginning.

You might have known the servant's arm can span EXACTLY K people.

For a fixed loop, such as 0, K, 2K mod N, 3K mod N, etc.

Enumerate the number of goblets putting together on 0 and K at servant's first visit, from -1000 to 1000.

Then the number of visits for (K,2K mod N), (2K mod N, 3K mod N) are easy to be counted.

And there can be many loops, you should process them one by one.

1276 - The Train

This algorithm is thought up by my former teammate Lin Tao.

First write them in the form like AB BA AB BA (enumerating the length), then insert AA and BB between them.

It's a combinatorial problem then.

Pay attention to that special case: all-A and all-B.

1277 - Cops and Thieves

A hateful problem:).

Split one point i into two (i and i'), the capability of i->i' is the number of policemen needed.

The capability of others is infinity.

We can prove the maximum-flow (minimum cut) is just the answer.

There is a bug in problem description (I don't know whether it has been deleted), it says you can't put police in the starting and ending point. This is obvious wrong. Or else, the sample will be impossible to be understood.

1278 - “… Connecting People”

Tooooooooo difficult to understand.. I'll help you understand it well.

It means, if we're processing the ith sentence "i: CALL X" (i,X in 0~99)

The program will push sentence i+1 into STACK, then process sentence X.

For BELL&RET command, it plays buzzer for 1 unit of time, then pop the STACK, and process that sentence.

It's easy to see, that we must use something about binary, coz there will be at most 10^9 units of time.

Let's see this program:

CALL 1

CALL 2

BELL&RET

It plays for 4 units of time

CALL 1

CALL 2

CALL 3

BELL&RET

It plays for 8 units of time

CALL 4

CALL 2

CALL 3

CALL 4

BELL&RET

It plays for 9 units of time

...

Let's analysis it:

i: CALL i+1 Process the (i+1)th sentence twice

we'll name it "CALL NEXT" below

i: CALL BELL&RET (call the sentence where BELL&RET is) Play for 1 unit of time.

The remaining puzzle is to make them up. Regard "PLAY X" is a set of commands which play for X units of time.

It's easy to find out a binary algorithm

PLAY X =

{

CALL BELL&RET

CALL NEXT

PLAY X div 2

} when X mod 2 = 1

PLAY X =

{

CALL NEXT

PLAY X div 2

} when X mod 2 = 0

Notice PLAY 1 =

{

BELL&RET

}

Then it's easy to be achieved by recurrence. Right?

1279 - Warehouse

10^7 was written as "107" in the past, that spent me a lot of time during the contest.

For saving time, I wrote a program which use binary search to determine the minimum number.

Coz the height of goods are all lower than 1000, so I use a hash array 1..1000 to record them.

I don't know why, I must use "readln" in c/c++ to get AC.. sigh~

1280 - Topological Sorting

The easiest problem among this contest. Number them by the given order, and check it!

1281 - River Basin

First join the rivers as much as possible, then count the convex polygon of them one by one.

1282 - Game Tree

Using the easiest gaming strategy, if new states are all win states, then this one is lose. Or else this one is win state.

1283 - Dwarf

Use logarithm to change “multi” to “sum”.

1284 - Cosmic Poker

Problem description: (this might help you understand it well)

Given N(N<20) cards, value 1~100, suit 1~10. Given a starting state, you’re asked to make them in order:

1. The cards with the same suit must be put continuously.

2. The odd-suit (blue) cards and even-suit (yellow) cards must be alternate.
(It is guaranteed that the | number of odd suits - number of even suits | <= 1)

3. Cards with the same suit must be put in the ascending or in descending order. All suits must be ordered in the same way.

Now you must order them in the minimal number of movements. (You can take one card out and put it in any position in a single movement)

The 1st testdata is

5

10 1

12 2

8 2

4 4

7 4

Output:

2

(I tested a lot, finally got it :)

Algorithm:

In the optimized way to order them, each card will be moved at most once. The cards that don't need to be moved formed a "legal list". The "legal list" is just a list of cards that is legal by the above "1st" and "3rd" conditions.

Such as this example:

7

1 3

2 3

3 3

1 1

2 1

3 1

2 2

The first 6 cards formed a “legal list”, and we can move the last card to the middle. Till now, have you thought of an idea?

We can use $O(2^n)$ to enumerate a “legal list”, then check whether the remaining cards can help match “2nd” condition, that is: if the same-color-suit cards are continuous, then there must be enough cards with different color (out of “legal list”) to be moved between them.

1285 - Thread in a hyperspace

I don’t know how to prove, but it is really obvious that we can use vector to calculate inclination between any two lines, and the module of a vector is just two points’ distance. Put them onto a single plane, then 2-dimension analytic geometry will help you solve it.

1286 - Starship travel

First we should get the Greatest Common Divisor of p and q. If n and m is not its multiple, it must be no solution.

Now we write N and M as px+qy. Suppose N=x1*p+y1*q, M=x2*p+y2*q. Write them as following:

x1 y1

x2 y2

Now our aim is to make them all to ZERO. We can have the following operations:

1. (+q,-p)+(+q,+p) = (+2q,0), x1, y1, x2, y2 can be added or subtracted by 2. So we can write them as 0 or 1.

2. (+p,+q), so we can do inc(x1),inc(y2) or inc(x2),inc(y1).

3. Redenote them. Inc(x1,q), inc(y1,p) or the symmetrical ones.

Now we can get the necessary sufficient condition of “no solution”:

p mod 2 = q mod 2 = 1, and (x1+y2) <> (x2+y1) mod 2.

1287 - Mars canals

Scan the graph once, with only O(N) in space, you can discover the longest chain of ‘S’ and ‘s’.

1288 - Astrolocation

Read the problem CLEARLY! The main idea of it is to ask you to find the maximum of a sequence.

1289 - One way ticket

Use f[i,j] to denote the total of i-digit numbers whose digit root is j. It’s very easy to calculate them.

You can find orderliness in it:D

1290 - Sabotage

Sort them in descending order!

1291 - Gear-wheels

You can construct a tree by the connections between gears. Then it’s a DFS problem.

1292 - Mars space stations

We’re asked to count $F^{(N-1)}(K) - L$. Notice that F(153) = 153. This is a black hole which will be got into by many of the numbers. Just simulate as you like~~~

1293 - Eniya

Writeln(N*A*B*2);…….@#(*$&@#(*%^!&#*(%^()#!*%

1294 - Mars satellites

Coz $ANGLE_{CAD}=ANGLE_{CBD}$, so we can have two equations, calculate out cos(CAD) correctly.

Now the length of CD is here. Notice that there’s a big bug in the testdata, if coz(CAD)>1, you MUSTN’T output “impossible”…. (I don’t know whether this bug is fixed already)

1295 - Crazy notions

Compute the first 500 answers, you'll see that the answer is among 0,1 and 2.

By the Fermat-Euler lemma, we can see that the answer for n and (n mod 2000) is the same.

In fact, using (n mod 100) is okay..

1296 - Hyperjump

Use a variable SUM to record the sum currently. From 1st number, if SUM<0, SUM:=0. Or else SUM:=SUM+currentnum.

And use another variable MAXSUM to record the maximal SUM. That is just the answer.

I think this algorithm is easy but fantastic, O(N) in time, and O(1) in space :D

1297 - Palindrome

Don't be trapped in DP. The best solution is just enumerating! Enumerate the middle number, then check whether it is correct.

1298 - The knight

N=2,3,4, no answer. We can only use search to solve this problem. When n=8, your program might get TLE. A little trick is here, consider the points whose degree is 1:D

1299 - Psylonians

Simulating problem. Your program will be made up by "if"s…

The most difficult part is the understanding for rotating.

The robot can move backward, so it's a little more difficult to find the minimal turn angle. E.g:

The target is on -150º, you can turn the robot +30º, then move straight backward to the target.

Enjoy it:D