





**热烈欢迎来到参加
上海GDG社区交流分享活动！**

Content

- Introduction to Code Jam
- Registration
- Rounds
- Problem example
- Other resource
- Code Jam History

What is Code Jam?

- Code Jam continues to bring together professional and student programmers from all over the world to solve tough algorithmic puzzles.

code jam

```
cout << "hello, world!" << endl;
```

What is Code Jam?

- Began in 2003, to identify top engineering talent for potential employment at Google.
- In 2008, 11000 people competed
- In 2009, 8605 people competed
- In 2010, 10000 people competed
- In 2011, 20000 people competed
- In 2012, 35000 people competed

Who is eligible?

- 13yr +
- Not a current employee/intern of Google
- Not a immediate family member of a Google employee
- Not a resident of anywhere that the contest is prohibited by law.

Rounds

- **Qualification Round:** lasts 25hours.
- **First Round:** Round 1A, Round 1B and Round 1C. Each lasts for 2.5 hours. Top 1000 -> Round 2.
- **Round 2:** 3000 contestants, lasts 2.5 hours. Top 1000 -> T-shirt, top 500 -> Round 3.
- **Round 3:** 500 contestants, lasts 2.5 hours. Top 25 -> Onsite Finals.
- **Onsite Finals:** Travel to a Google office

Code Jam T-shirt 2011

**WHAT LANGUAGE
DO YOU SPEAK?**

```

sum      [ d!
(s, s') *
choose
(n - s - 1) (s - s' - 1)
for lim := len(list);          lim
s/(\(7\)<(\,*\))(>.*\1\,\,\)/4\2
fold_gcd([H|T], X) :- fold_gcd(T, Y)
line= cat $1 | cut -c $1 | tr -d '\n'
N { K 2 mod 0 eq { /good false def } if /K K
D = length(list:filter(fun(V) ->
cross_wire(V, W) end, Sofar))
(for [dir
IZ 0
MMXI?
BIG LIKE
(for
x = (+ i
(random
(* 2.0d0 r))
foreach my $y
(keys %{$M->{$
list($R, $k, $n) =
explode(' ', trim
[<=(x:long, y:integer)
: boolean -> not
of [SOME n, SOME
k] => spawn ((c,n,k)
divl %ecx addb
$'0' %dl decl %edi
movb %dl, (%edi)
cmpl $0, %eax jnz it
s/(\
def      diff = times[1..-1].collect
{ def result = it - start; start = it;
: k2**n-1? ( a n - ? ) [ 1 + ] [ 2^ ] bi*
Case #'c':c@-~2\?{.@&=: ON": OFF'if]
48~^[ $10=-][48-~10*+^]#%\[ $0=-][1-~$2
/$2*@[ @%0@%0@?]\#%$0$0=["OFF"]
while (grup < n && ludi + g[(c+grup)%n] <= k) { ludi +
writefln("Case #%s: %s", tc, ((K + 1) & ((1 << N) - 1))) ? "OFF" : "ON");
(defun purep (n lst) (or (= n 1) (let ((p (position n lst))) (and
if a.(x).[y] = c then ( incr m; if !m = k th en raise Fo un
for(i=2, matsize
c then
raise Found )
2 idiv def } repeat
(fgets($in)))
(1+ p) lst)
iter
if a.(x).[y]
k then
s' <- [1
M->{$x})} {
: k2**n-1? ( a n
(lo ng!(y) < x)]
[ 1 + ] [ 2^ ] bi* m
od zero?;
gcd(H, Y, X)
result }.sort
0:c;[-](:2/[['
g[(c+gru
p)%n];
code jam
print "hello, world!"

```

Problem example

code jam

`System.out.println("hello, world!");`

Qualification Round 2011

A. Bot Trust

[B. Magicka](#)

[C. Candy Splitting](#)

[D. GoroSort](#)

[Contest Analysis](#)

[Ask a question](#)

[View my submissions](#)

Submissions

Bot Trust

10pt **Correct**
10560/12572 users
correct (84%)

10pt **Correct**
10291/10514 users
correct (98%)

Magicka

10pt **Correct**
8886/10218 users
correct (87%)

15pt **Correct**

Practice Mode Rank: 685 Score: 100

Problem A. Bot Trust

This contest is open for practice. You can try every problem as many times as you like, though we won't keep

Small input
10 points

Solve A-small

Large input
10 points

Solve A-large

Problem

Blue and Orange are friendly robots. An evil computer mastermind has locked them up in separate hallways to test them, and then possibly give them cake.

Each hallway contains 100 buttons labeled with the positive integers $\{1, 2, \dots, 100\}$. Button k is always k meters from the start of the hallway, and the robots both begin at button 1. Over the period of one second, a robot can walk one meter in either direction, or it can press the button at its position once, or it can stay at its position and not press the button. To complete the test, the robots need to push a certain sequence of buttons in a certain order. Both robots know the full sequence in advance. How fast can they complete it?

For example, let's consider the following button sequence:

O 2, B 1, B 2, O 4

Here, O 2 means button 2 in Orange's hallway, B 1 means button 1 in Blue's hallway, and so on. The robots can push this sequence of buttons in 6 seconds using the strategy shown below:

<https://code.google.com/codejam/contest/975485/dashboard>

Other resource

- TopCoder www.topcoder.com/tc
- ACM ICPC
- SPOJ www.spoj.pl

Code Jam History



ACRush

IOI 2004, 3rd Place

A* Star 2005 1st Place

A* Star 2006 1st Place

TopCoder 1st Place

ACM/ICPC Global 2nd Place

Code Jam 2008, 2009 1st Place



开放 分享 创新

developers.google.com