

全国信息学奥林匹克竞赛(NOIP2018)模拟赛

提高组

(选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	平抛	数字虫	最小路径
英文题目与子目录名	horizontal	bug	path
可执行文件名	horizontal	bug	path
输入文件名	horizontal.in	bug.in	path.in
输出文件名	horizontal.out	bug.out	path.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	20	20	20
每个测试点分值	5	5	5
附加样例文件	有	有	有
结果比较方式	浮点数比较 (Special Judge)		全文比较 (过滤文末空格及行尾空格)
题目类型	传统	传统	传统
运行内存上限	256MB	256MB	256MB

二、提交源程序文件名

对于 C++语言	horizontal.cpp	bug.cpp	path.cpp
对于 C 语言	horizontal.c	bug.c	path.c
对于 pascal 语言	horizontal.pas	bug.pas	path.pas

三、编译命令 (不含任何优化开关)

对于 C++语言	g++ -o horizontal horizontal.cpp -lm	g++ -o bug bug.cpp -lm	g++ -o path path.cpp -lm
对于 C 语言	gcc -o horizontal horizontal.cpp -lm	gcc -o bug bug.cpp -lm	gcc -o path path.cpp -lm
对于 pascal 语言	fpc horizontal.pas	fpc bug.pas	fpc path.pas

注意事项:

1. 文件名 (程序名和输入出文件) 必须使用英文小写。
2. C/C++中函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时的返回值必须是 0。
3. 全校统一测评采用的机器配置为: CPU Intel Core i5-7200U @2.71GHz, 内存 4GB, 上述时限以此配置为准
4. 只提供 Linux 格式附加样例文件。
5. 特别提示: 测评在 NOI Linux 下进行。

1.平抛

(horizontal.cpp/c/pas)

【问题描述】

一质点从一定高度水平抛出，经过一系列圆形挡板的干扰，最终落在低地面上。挡板分为两种类型：类型 1 只改变质点的速度方向，不改变质点的速度大小；类型 2 只改变质点的速度大小，不改变质点的速度方向。当质点与挡板接触时，其竖直方向上的分速度将立即变为零，水平方向上的分速度不变；与此同时，挡板将立即移动，使圆心位于质点与挡板接触的位置。你需要计算质点落地（质点的高度等于零时落地）时的位置、速度。

【输入】

第一行，三个实数 x_0 、 y_0 、 z_0 ，表示质点的初始位置。 $-1000 \leq x_0, y_0 \leq 1000$ ， $0 < z_0 \leq 1000$

第二行，两个实数 v_x 、 v_y ，表示质点初始时在 x 、 y 方向上的分速度。 $-1000 \leq v_x, v_y \leq 1000$ 。

第三行，一个整数 n ，表示挡板的数量。

接下来 n 行，每行包含四个实数 x 、 y 、 z 、 r ，分别表示挡板的位置、半径；之后是一个整数 t ，表示挡板的类型，之后是一个实数 d ，当 $t=1$ 时， d 表示速度方向的该变量， $d \in (0, 2\pi)$ ；当 $t=2$ 时， d 表示速度大小的该变量， $d \in [-10^3, 10^3]$ 。保证所有挡板不相交。 $-1000 \leq x, y, r \leq 1000$ ， $0 < z \leq 1000$ 。

【输出】

第一行，两个实数，表示质点落地的位置（ x, y ）。

第二行，三个实数，表示质点落地时的速度（用 x, y, z 方向的分速度表示，负号表示与坐标轴正方向相反）。

如果小球在某一瞬间速度小于 10^{-3} ，则只输出“Error”（不含引号）。

【输入输出样例】

horizontal1.in	horizontal1.out
0.0 0.0 10.0 1.0 0.0 3 2.0 0.0 5.1 3.0 1 3.141592654 3.2 5.0 5.2 0.3 2 -5.0 -3.0 0.0 0.2 1.0 2 -2.0	-1.7979695 -0.0000000 1.0000000 0.0000000 1.9798990
horizontal2.in（见选手目录）	horizontal2.out（见选手目录）
horizontal3.in（见选手目录）	horizontal3.out（见选手目录）

【子任务】

测试点	n	其他性质 1	其他性质 2
1	0	无	有
2	0	无	无
3	1	有	有
4	1	有	无
5	≤ 50	有	有
6~9	≤ 500	有	有
10~12	≤ 1000	有	有
13、14	≤ 10000	无	有

15、16	≤ 10000	无	无
17~20	≤ 100000	无	无

其他性质 1：保证各个挡板的高度（ z 坐标）不同

其他性质 2：保证质点的 y 坐标始终为 0

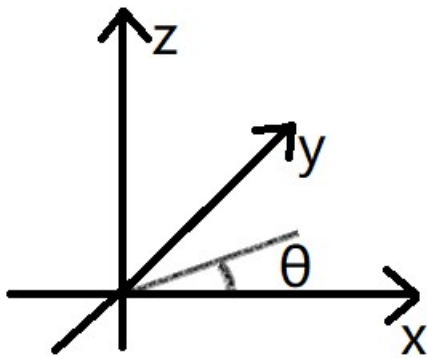
对于每一个测试点：

与答案的差的绝对值（按误差最大的数字计算）	得分
$\leq 10^{-3}$	5
$\leq 10^{-2}$	4.5
$\leq 10^{-1}$	3
≤ 1	1
> 1	0

【提示】

重力加速度 $g=9.8\text{m/s}^2$

题目中的坐标系如图所示：



2.数字虫

(bug.cpp/c/pas)

【问题描述】

在一个广场上散布着一些数字。每个数字由 1~5 组成，且小于 10^{500} 。一只小虫从其中任意一个数字出发，途径所有数字，最终回到原来的数字。小虫每从一个数字爬行到另一个数字都会有一定代价，收获的大小被定义为两个数字的相似度除以它们间的距离，除起点外，同一个数字不能重复经过。请你计算小虫获得最小的代价。

为了计算两个数字的相似度，我们添加若干个 0 使两个数字长度相等，在每一格位置，两个数字匹配的相似度如下表所示：

	0	1	2	3	4	5
0	-100	-1	-2	-8	-16	-32
1	-1	6	-2	-4	-1	-2
2	-2	-2	7	2	3	-7
3	-8	-4	2	8	1	5
4	-16	-1	3	1	9	2
5	-32	-2	-7	5	2	15

按照这种规则 123、32 可以变化为 123、302，相似度 $=-4-2+2=-4$ ，当然也可以变化为 123、032，这样计算出的相似度为 3，或者变化为 1230、0032，算得的相似度同样是 3。事实上，其他变化得到的相似度均不大于 3。由于同样两个数字经过不同的变化得到的相似度不同，我们规定，两个数字的相似度为所有变化中最大的相似度。这样，两个由 1~5 组成的数字的相似度是唯一确定的。我们还规定，任何两个数字之间的相似度大于等于 0，于是 1、2 之间的相似度等于 0，而不是-1。

【输入】

第一行，一个整数 n ，表示数字的个数。

接下来 n 行，每行包含一个整数 m 和两个实数 x, y ，表示数字的值和数字的位置，保证两个数字不在同一个位置。

【输出】

一个浮点数，表示小虫的最小代价。

【输入输出样例】

bug1.in	bug1.out
3 123 0.0 0.0 32 0.0 1.0 123 1.0 0.0	26.1213203436
bug2.in	bug2.out
5 135 1.3 43.2 23213 -32.1 0.123 5212 -1.53 9.0 5555 99.9 -99.9 1111111 0.23 -89.32	0.0291401702
bug3.in (见选手目录)	bug3.out (见选手目录)
bug4.in (见选手目录)	bug4.out (见选手目录)

样例 1 说明：

第 1、2 个数字间的收获为 $3 \div 1=3$

第 2、3 个数字间的收获为 $3 \div \sqrt{2}=2.1213203436$

第 1、3 个数字间的收获为 $21 \div 1=21$

【子任务】

测试点	n	m	x,y
1、2	≤ 3	$\leq 10^{10}$	$-100 \leq x,y \leq 100$
3、4、5	≤ 5		
6、7	≤ 3	$\leq 10^{30}$	
8、9、10	≤ 5	$\leq 10^{30}$	
11、12	≤ 10	$\leq 10^{10}$	
13、14、15	≤ 10	$\leq 10^{100}$	
16~20	≤ 16	$\leq 10^{500}$	

对于每一个测试点：

与答案的差的绝对值	得分
$\leq 10^{-6}$	5
$\leq 10^{-5}$	4.5
$\leq 10^{-4}$	4
$\leq 10^{-3}$	3
$\leq 10^{-2}$	2
$\leq 10^{-1}$	1
$> 10^{-1}$	0

3.最小路径

(path.cpp/c/pas)

【问题描述】

我们非常熟悉图论中的“最短路径”，这里的最短的意思是边的权重和最小，假如我们改变一下条件，规定两个点之间的“最小路径”为两个点之间的所有路径中，边的权重的最大值最小的路径。

给定一张图和一系列询问，对于每一个询问，请回答询问中的两个点之间的最小路径。

【输入】

第一行，两个整数，n、p，表示节点的数量和边的数量。

接下来 p 行，每行包含三个整数，s、t、r，表示点 s、点 t 之间存在一条权重为 r 的无向边。

第 p+2 行，一个整数 q，表示询问的个数。

接下来 q 行，每行有两个整数 f、g，表示询问点 f、点 g 间的最小路径。

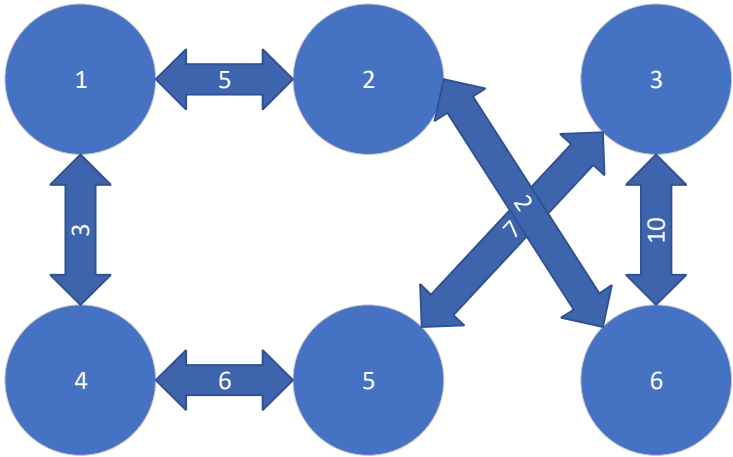
【输出】

对于每一条询问，输出一行，含一个整数，最小路径中权重最大的边的大小。如果两个点之间无法互相到达，输出-1。

【输入输出样例】

path1.in	path1.out
6 6 1 2 5 1 4 3 5 4 6 2 6 2 5 3 7 3 6 10 3 1 5 5 1 6 4	6 6 5
path2.in (见选手目录)	path2.out (见选手目录)
path3.in (见选手目录)	path3.out (见选手目录)

样例说明：



询问一：1 -> 4 -> 5

询问二：5 -> 4 -> 1

询问三：6 -> 2 -> 1 -> 4

【子任务】

测试点	n	p	q	其他约束条件
1、2	≤ 100	≤ 200	≤ 1000	无
3~5	≤ 10000	≤ 50000	≤ 10	无
6、7	≤ 1000	≤ 2000	≤ 100	无
8、9	≤ 10000	≤ 50000	≤ 30000	图退化成一条链
10、11	≤ 10000	≤ 50000	≤ 1000	图中存在某个节点，与其他节点通过一条边直接相连
12~15	≤ 10000	≤ 50000	≤ 10000	保证图是联通的
16~20	≤ 10000	≤ 50000	≤ 30000	无

$1 \leq r \leq 10^8$ 。