

PROGRAMMER_1024_2021

2021年可兰达程序员节：数据挖宝

活动流程

1. 2021年10月8日：宣讲会。
2. 2021年10月9日下班前：完成组队，提交队伍成员和队名。
3. 2021年10月8日-10月22日下午3点前：开发调试阶段
4. 2021年10月22日下午3点-4点：提交程序到指定位置
5. 2021年10月22日下午5点：现场抽签决定分组，然后是比赛环节，比赛为淘汰赛，一局定胜负，胜出者进入下一次抽签分组直至决出前三名，每组最多3个退伍。
6. 2021年10月29日下午5点：颁奖以及前三名的分享会。

游戏规则

- 组委会提供128个文件，每个数据文件包含200w条左右数据，每条数据用\n换行符分割
- 每一条的数据格式为json，形如：

```
{"locationid":"2ixewekdyvgbfmzli9iifm1w9hnd2ij5kr1avy1zw3c7rl","magic":"9248642713483188"}
```
- 如果locationid代表的字符，去掉字母部分，剩下的数据部分，通过加/减/乘/取余 1024，恰好等于 magic 字符串所代表的数字，那么这个地点就表示存在宝藏。

例如：{"locationid":"2mnab0kquw4uuu8nnm","magic":"2"} 抽取后的数字为2048， $2048/1024 = 2$ 恰好等于magic代表的数字，所以这个地点就代表有宝藏。

- 找到宝藏地点后，需要把对应的locationid通过post请求，发送到我们的dig接口，如果该宝藏还未被其他队伍找到，那么获得1分.
- 对找到的所有magic，对其中n个数($n \geq 4$)，若通过加减乘除（可以有"()”）等于1024，那么把对应的计算方法，生成算式，发送到我们的formula接口，如果对应的n个数字的算式还没有被其他队伍发现，获得 $n \times n$ 分，算式中的数字用locationid代表.

例如：

(vbncooczywr3yqbju8bo6uq5xcmq93ihd6b24dh5gacskeiubdwlyzckocd0z5k+gp16z0epmyodmqc5hywuidodsfqhd816w7uze1xz2jlc00rddaf5267m5fxvdd0h)*

(1249eabfioy2cdro0ggsaqsovfadqxdnykmys2lke5xa2a3zjsxb8n0rqd2kyej-lzvlff6idy8e9w4oe14ufuz82inpv n64p1izh4jbzedpiurx9d8glqsp6b9i4phs)

- 最后，得分最多的队伍获胜，如果有队伍得分一样，那么首先获得第1分的队伍获胜.

说明：1 - locationid的长度为固定的64个字符.

2 - magic的长度为2-32个字符表示的数字.

3 - magic "027" 表示 27.

4 - 比赛当天抽两个组测试一下.

5 - 每场比赛时间3分钟.

奖项设置

第一名：奖金 4096 元

第二名：奖金 2048 元

第三名：奖金1024 元

参赛说明

1.参赛队伍请提供一个可在linux命令行下直接运行的可执行文件

2.数据目录和可执行文件在同一级目录

例如：

队伍 a 提交了他们的可执行文件 a.out

比赛时，会以 ./a.out 来运行，data目录在a.out的同级目录，运行参数可自行指定

./data/Treasure_0.data

./data/Treasure_1.data

./data/Treasure_2.data

...

./data/Treasure_1023.data

./a.out

机器及环境

机器配置 - 阿里云计算型c6

| 规格族 ① | 实例规格 | vCPU ② | 内存 ③ | 处理器主频/睿频 | 内网带宽 ④ | 内网收发包 ⑤ ⑥ | 存储IOPS 基准/峰值 ⑦ | 参考价格 ⑧ ⑨ | 处理器型号 |
|--------|---------------|--------|-------|-----------------|-----------|-----------|-------------------|-----------|--|
| 计算型 c6 | ecs.c6.xlarge | 4 vCPU | 8 GiB | 2.5 GHz/3.2 GHz | 最高 5 Gbps | 50 万 PPS | 2.1 万/- | ¥ 0.78 /时 | Intel Xeon(Cascade Lake) Platinum 8269CY |

操作系统 - Ubuntu 18.04 64位

会准备一个调试环境给大家，详细硬件参数和规格可自行登陆查看

API

接口地址：<http://47.104.220.230/dig>

请求方法：POST

请求Header: application/json

请求参数:

```
{
  "locationid":"2ixewekdyvgbfmfzli9iifm1w9hnd2ij5kr1avy1zw3c7r1", //宝藏地点id
  "token":"oaksuquwqiw=928182ijasj" //请求token
}
```

响应数据:

```
{
  "errorno":0, // 0表示dig成功, 获得1分; 1表示该id不是宝地点的id, 不得分; 2表示该id的宝
  藏已经被挖了, 不得分;-1表示其他错误;
  "msg":"game has end." //当errno为-1时, 会包含该字段
}
```

接口地址: <http://47.104.220.230/formula>

请求方法: POST

请求Header: application/json

请求参数:

```
{
  "formula":"(qwe+mas)*su5mq-wpqs*yyq21hs", //算式字符串
  "token":"oaksuquwqiw=928182ijasj" //请求token
}
```

响应数据:

```
{
  "errorno":0, // 0表示成功, 其他返回值表示错误;
}
```

POST请求示例

curl示例:

```
curl -d '{"token":"test1","locationid":"moaooosas"}' http://47.104.220.230/dig
```

go语言:

```
data := make(map[string]interface{})
data["locationid"] = item.Locationid
data["token"] = "test1"

bytesData, _ := json.Marshal(data)
resp, _ := http.Post("http://47.104.220.230/dig", "application/json",
bytes.NewReader(bytesData))
defer resp.Body.Close()
body, _ := ioutil.ReadAll(resp.Body)
fmt.Println(string(body))
```