

# New problem to be configured

---

## 题目大意

给一种 HackVM 语言，要求使用该语言实现一些任务。

## 分点分析

### 测试点 1

考察选手读题能力。代码为 `rr+p`。

### 测试点 2

考察选手读题能力。代码为 `rr1^1^/*-p`。

### 测试点 3

快速幂。为了防止在 HackVM 里写循环，可以使用 C++ 输出 32 次操作。

### 测试点 4~9

实现对应算法即可。

考虑到对应算法直接在 HackVM 中实现较为困难，且较难调试，可以实现一个转换器，来简化 HackVM 中的操作。

至少需要实现的部分有：`快速 push 大于 9 的数到栈中` 和 `计算 $ 和 g 的位置`。

实现完这两部分之后，应该能较容易的实现测试点 4、5。测试点 6、7 也可能可以实现。

考虑局部变量可以放在操作数栈中，如果进一步实现 `自动管理操作数栈中的变量` 和 `简化的实现条件判断与循环`，可以实现测试点 8、9。

### 测试点 10

假设现在操作数栈中有一个字符串。如果实现一个算法，能先输出 `在 HackVM 下将该字符串 Push 到操作数栈中的代码`，然后输出该字符串本身。那么把这个算法的字符串放入操作数栈中，然后运行这个算法，得到的就是 Quine 代码。

字符串格式可以是 `前面是字符串每一位的 ASCII 码，最后是这个字符串的长度`。这个算法中不能有 `c` 操作，因为会改变后面的位置。

### 测试点 11

可以使用 HackVM 自带的操作数栈模拟给定代码的操作数栈。内存和调用栈可以分别指定为某个地址开始的一片内存。

对于每种操作，可以用一个函数执行。如果将内存和调用栈的地址存到内存中一个  $\leq 9$  的位置，那么可以在几次操作的时间内实现这些函数。

最后是模拟代码的过程。当前指针可以存到内存中一个  $\leq 9$  的位置。代码需要开一片内存存放，但是这片内存的地址也可以存放到内存中一个  $\leq 9$  的位置。除此之外，每次判断并调用函数的过程可以不用依次判断，而是直接把对应函数的指针存放到该操作的 ASCII 码指向的内存位置，这样可以大幅减小常数。

## 总结

本题是一道构造计算机的提交答案题，综合考察了选手的算法能力和代码实现能力。相信这道题目能够给选手的 IOI 之路一个有力的援助！