

Lab04 - Baguenaudier

Task

Baguenaudier is a traditional Chinese folk intellectual toy made of metal wire and consisting of 9 circular rings. The rings are fitted onto a horizontal board or various frames and threaded through a handle.

Now, we want to know how to minimize the number of operations required to unlock Baguenaudier.

Actually, we want to solve n-rings problem.

Additional Information

Specifically, we have n rings numbered from 1 to n.

One operation involves choosing a ring, then putting on or removing from the board.

A ring which could be put on or removed from the board, needs to meet **one of the following conditions**:

1. It is the 1st ring.
2. It is the i-th ring, and the (i-1)-th is on the board, but **the first (i-2) rings (which means the rings numbered 1 to i-2, if they exist)** is not on the board.

Let $R(i)$ is the process of removing the first i rings from the board, $P(i)$ is the process of putting the first i rings on the board.

It's not difficult to find that the process of solving problems is **recursive**, besides $R(i)$ and $P(i)$ are **inverse processes** of each other.

For example,

$R(0) = \text{nothing to do}$, $R(1) = \text{remove the 1}^{st} \text{ ring}$,

$R(i) = R(i-2) + \text{remove the } i^{th} \text{ ring} + P(i-2) + R(i-1), i \geq 2$.

Coding Your Operation

We use n-bit binary to represent the states of n rings, **the lowest(rightmost) bit representing the 1-st ring**.

The bits which are 0, means that these rings are putting on the board. Otherwise, these rings have been removed from the board.

Set the portion exceeding n bits to 0.

Your Job

1. The value of n will be set manually in **x3100**, and **n is a positive integer not exceeding 12**.
2. You should store all rings' state after the **1-st** operation at **x3101**, store the state after the **2-nd** operation at **x3102**, and so on, till you finish your operations.

For ease of understanding, **we have provided a C language program without some implementation details as a reference**.

You can complete it yourself to provide an answer, but **there is no need to submit it**.

Example

r means remove, p means put.

Address	Memory	Explanation
x3100	0000_0000_0000_0011	$n=3$
x3101	0000_0000_0000_0001	r the 1st ring
x3102	0000_0000_0000_0101	r the 3rd ring
x3103	0000_0000_0000_0100	p the 1st ring
x3104	0000_0000_0000_0110	r the 2nd ring
x3105	0000_0000_0000_0111	r the 1st ring

Solving 3-rings problem needs 5 operations at least.

Attention

For this lab, you are required to use **assembly code**. Please adhere to the following guidelines:

1. Your program should start with `.ORIG x3000`.
2. Ensure your program ends with `.END`.
3. Your last instruction must be `TRAP x25 (HALT)`.

Your submission should be structured as shown below:

```
PB22*****_Name.zip
├─ PB22*****_Name_report.pdf
└─ lab4.asm
```

请在 bb 系统上分开提交

Your report should be structured into the following sections:

1. Purpose
2. Principles
3. Procedure (e.g. bugs or challenges you encountered and how to solve them)
4. Result