

## Does Putting N at the End of the Array Work?

Given the stack shown here,  
**what should the subroutine return?**

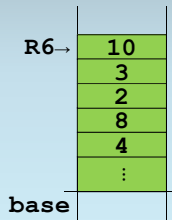
13? ( $N=2$ )

23? ( $N=4$ )

Something else? (Is  $N$  shown?)

The answer is ambiguous!

**(Such an approach is not acceptable.)**



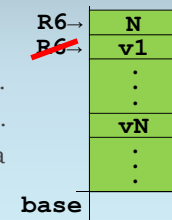
## One Other Solution is Possible

**How can the caller tell the subroutine the value of  $N$ ?**

1. Use a fixed value, such as 3.
2. Pass  $N$  in a register, say  $R2$ .
3. End the list with a non-data sentinel (such as -1).

But there is one more answer...

4. **Put  $N$  on top of the stack (always in a known position:  $M[R6]$ ).**



## A Stack for MP3

In MP3,

- you will use a stack
- to implement a depth-first search (DFS).

Given

- a list of extra events,
- each with several options for hour slot,
- you must try to find a combination
- that works without schedule conflicts.

## A Stack Frame Holds All Information for a Subroutine

Imagine that you are using

- an ISA with few/no registers, so
- you must use the stack to manage subroutine calls.

Let's **define a block of data**

- called a **stack frame** (or **activation record**)
- that **holds all** of the **information**
- needed **for one subroutine**.