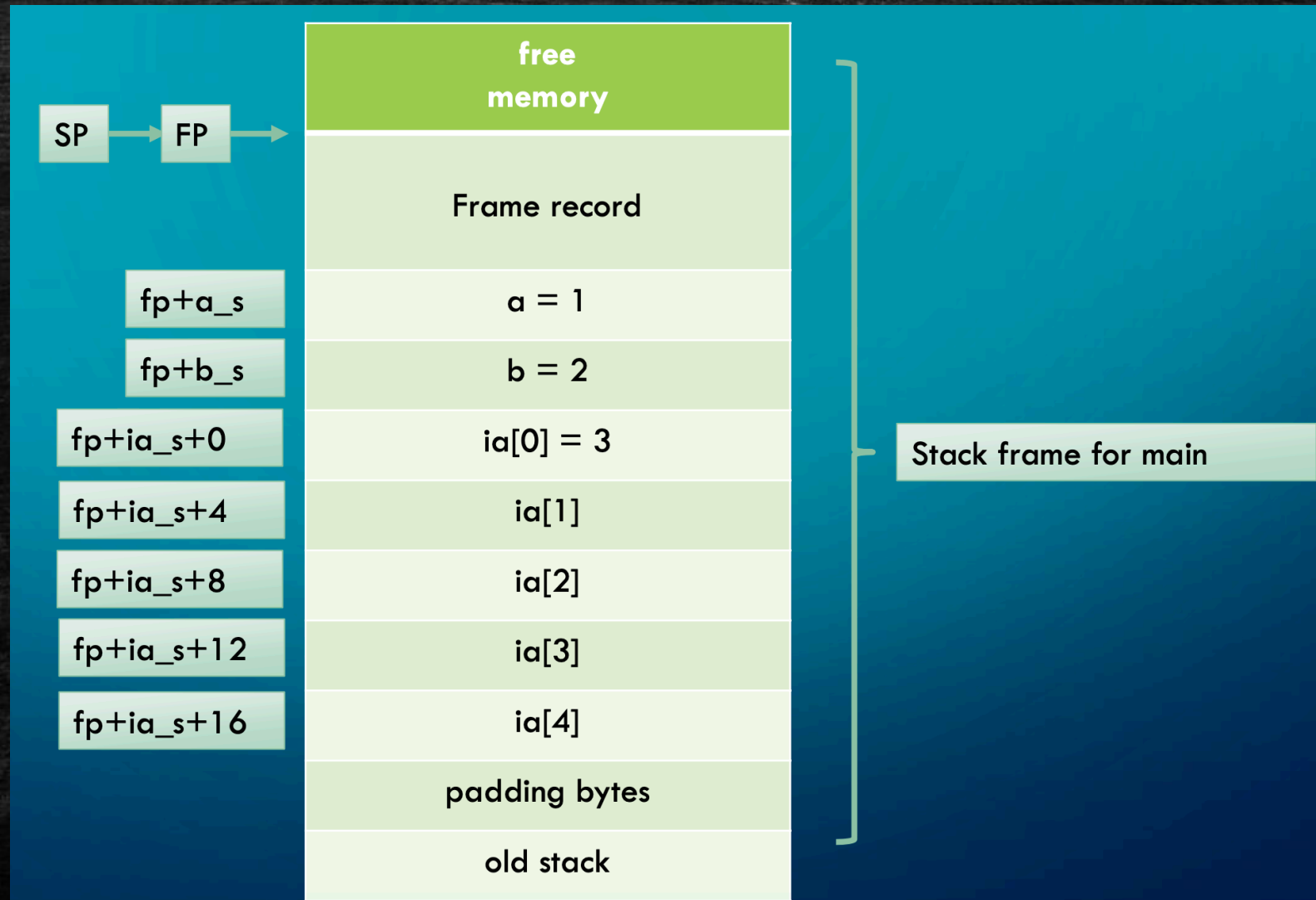


Memory used in stack for Example 1



Access elements of an array [1/2]

- Array elements are accessed using **load** and **store** instructions
- To simplify, set the Array Base Address = FP + offset_first_element
- The offset of element = index * element size
- e.g. store the 3rd element `ia[2] = 13` into stack (index = 2; element size = 4 bytes)

```
ia[2] = 13;
define(ia_base_r, x19)
define(index_r, x20)
define(offset_r, x21)
...
main: add ia_base_r, x29, ia_s // Calculate array base address: starting address of ia[0]
      mov index_r, 2 // set index to 2
      lsl offset_r, index, 2 // offset = i * element size, index << 2 = index * 4 (4bytes)

      mov w22, 13 // ia[2] = 13
      str w22, [ia_base_r, offset_r]
```


rand

- To generate random numbers we will use rand function
- Is a pseudo random number generator
 - Meaning: it will create the same sequence again and again every time program runs
 - e.g. if we are generating 5 random numbers in C with the help of rand() in a loop, then every time we compile and run the program our output must be the same sequence of numbers.
- Call same way as we call printf:
 - bl rand
 - The result is stored in w0
 - e.g. Generate a random number, mod the result with 256 and store in w20

```
bl rand          // generate a random number, result is in w0
and w20, w0, 0xFF // mod the result with 256
```