ABC by Example: Arrays (Top-Down Approach)

…and how to implement a simple stack

Stack Data Type

Provides two main operations:

- Push operation: Add an element
- Pop operation: Remove most recently added element

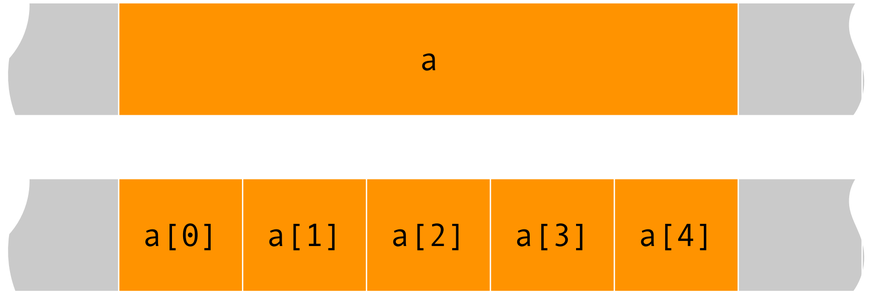
RAM:

```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    a[0] = 123;
    a[1] = 42;
    a[2] = 13;
    a[3] = 65;
    a[4] = 54;
}
```

```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    a[0] = 123;
    a[1] = 42;
    a[2] = 13;
    a[3] = 65;
    a[4] = 54;
}
```

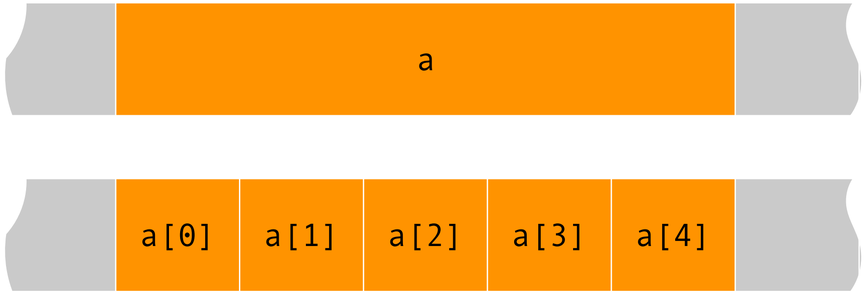
```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    a[0] = 123;
    a[1] = 42;
    a[2] = 13;
    a[3] = 65;
    a[4] = 54;
}
```

RAM:



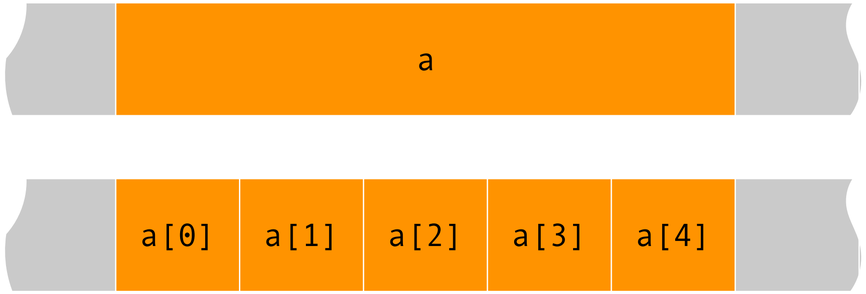
```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    a[0] = 123;
    a[1] = 42;
    a[2] = 13;
    a[3] = 65;
    a[4] = 54;

    for (local i: int = 0; i < 5; ++i) {
        printf("a[%d] = %d\n", i, a[i]);
    }
}
```

RAM:

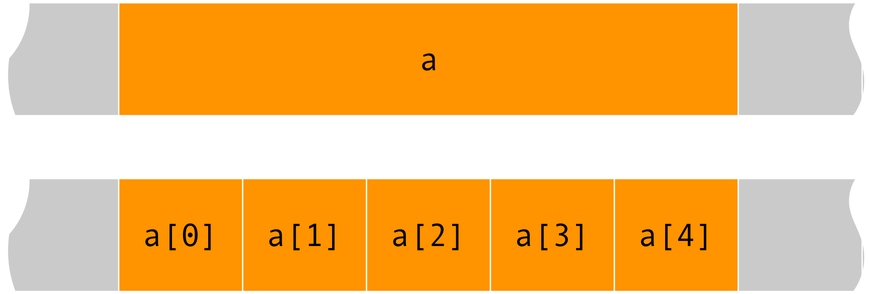


```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    a[0] = 123;
    a[1] = 42;
    a[2] = 13;
    a[3] = 65;
    a[4] = 54;

    for (local i: int = 0; i < 5; ++i) {
        printf("a[%d] = %d\n", i, a[i]);
    }
}
```

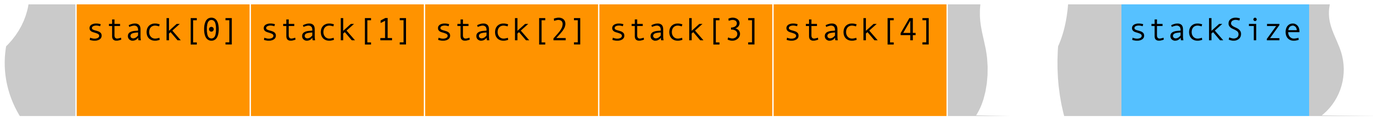
RAM:



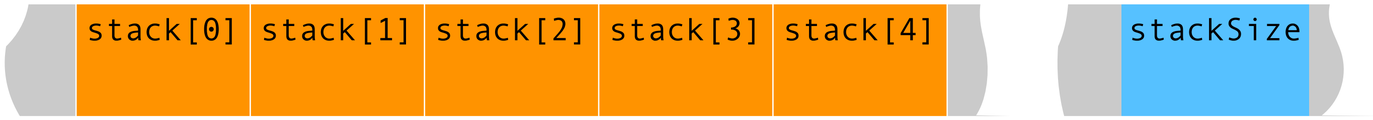
```
@ <stdio.hdr>

global a: array[5] of int;

fn main()
{
    printf("sizeof(a) = %zu\n", sizeof(a));
    printf("sizeof(a[0]) = %zu\n", sizeof(a[0]));

    printf("sizeof(a) / sizeof(a[0]) = %zu\n", sizeof(a) / sizeof(a[0]));
}
```



```
@ <stdio.hdr>

global
    stack: array[5] of int,
    stackSize: int = 0;

fn push(val: int)
{
    stack[stackSize] = val;
    ++stackSize;
}

fn pop(): int
{
    --stackSize;
    return stack[stackSize];
}

fn main()
{
    push(42);
    push(12);
    printf("pop returned: %d\n", pop());
    printf("pop returned: %d\n", pop());
}
```



```
@ <stdio.hdr>

global
    stack: array[5] of int,
    stackSize: int = 0;

fn push(val: int)
{
    stack[stackSize++] = val;
}

fn pop(): int
{
    return stack[--stackSize];
}

fn main()
{
    push(42);
    push(12);
    printf("pop returned: %d\n", pop());
    printf("pop returned: %d\n", pop());
}
```



```
@ <stdio.hdr>

fn main()
{
    local a: int;
    local b: int;

    a = 1;
    b = ++a;
    printf("a = %d, b = %d\n", a, b);

    b = a++;
    printf("a = %d, b = %d\n", a, b);
}
```

```
@ <stdio.hdr>

global
    stack: array[5] of int,
    stackSize: int = 0;

fn push(val: int)
{
    assert(stackSize < sizeof(stack) / sizeof(stack[0]));
    stack[stackSize++] = val;
}

fn pop(): int
{
    assert(stackSize > 0);
    return stack[--stackSize];
}

fn main()
{
    push(42);
    push(12);
    printf("pop returned: %d\n", pop());
    printf("pop returned: %d\n", pop());
}
```