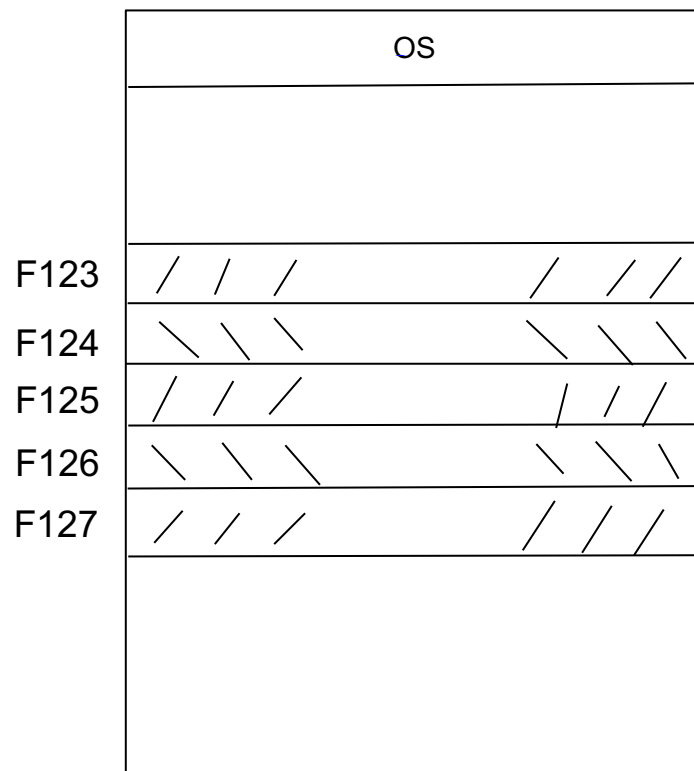
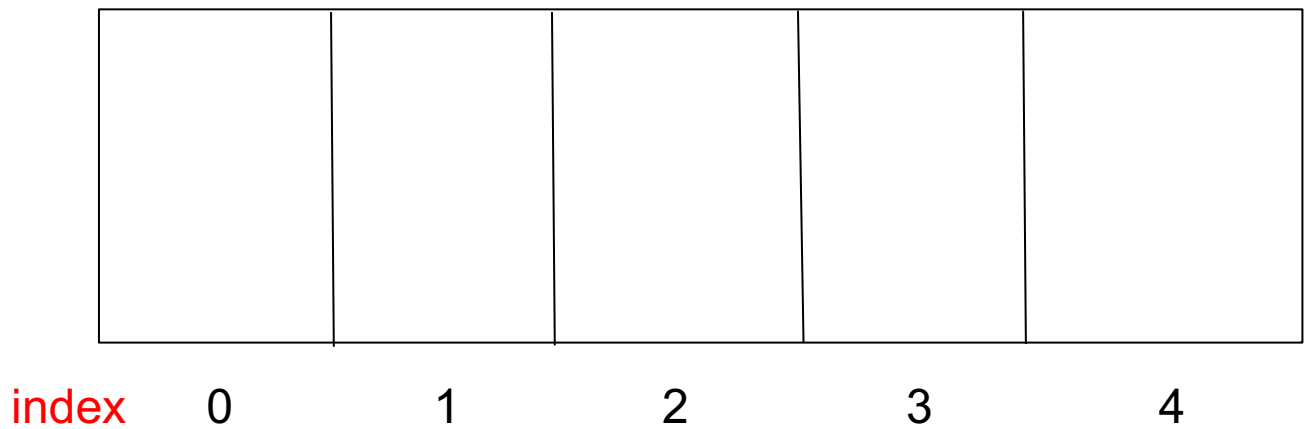


C Programming

Pointers and Arrays

Let's use the following array:

```
int a[5];
```



The elements of this array are:

a[0]

a[1]

a[2]

a[3]

a[4]

In C, the name of an array is the same as the memory address location of the first element.
i.e.,

$$a \approx \&a[0]$$

```
/*  
Program to show that the name of an array is the same as the address of  
the first element  
*/  
  
#include <stdio.h>  
  
#define SIZE 5  
  
int main()  
{  
    int a[SIZE];  
  
    printf("a is %p and &a[0] is %p", a, &a[0]);  
  
    return 0;  
  
} // end main()
```

Repl 10.1: <https://replit.com/@michaelTUDublin/101-Pointers-and-Arrays-p1#main.c>

Given that the name of an array is the same as the memory address location of the first element, then what is equivalent for the other elements

`a` \approx `&a[0];`

`a + 1` \approx `&a[1];`

`a + 2` \approx `&a[2];`

`a + 3` \approx `&a[3];`

`a + 4` \approx `&a[4];`

Assuming the above, let's try the following:

`*a` \approx `a[0];`

`*(a + 1)` \approx `a[1];`

`*(a + 2)` \approx `a[2];`

`*(a + 3)` \approx `a[3];`

`*(a + 4)` \approx `a[4];`

Let's look at the following code to explain this:

```
/*  
Program to show how to access an array using the dereference operator  
*/  
  
#include <stdio.h>  
  
#define SIZE 5  
  
int main()  
{  
    int a[SIZE] = {2, 4, 6, 8, 10};  
    int i;
```

```

// Display the contents of the array
for(i = 0; i < SIZE; i++)
{
    printf("\nElement %d contains %d, same as %d", i, a[i], *(a + i));
} // end for

return 0;

} // end main()

```

Repl 10.2: <https://replit.com/@michaelTUDublin/102-Pointers-and-Arrays-p2#main.c>

To summarise the above, you now have **TWO** ways to access an array:

1. **Subscript notation**

This is when you use [and] to access the elements of an array
e.g., a[0], a[1], etc.,

2. **Pointer notation**

This is when you use the dereference operator to access the array
e.g., *a, *(a + 1), *(a + 2), etc.,

$$\text{array_name}[i] \approx *(array_name + i)$$

Subscript notation

Pointer notation

```

/*
Program to show how to access an array using subscript notation and
pointer notation
*/

#include <stdio.h>

#define SIZE 5

int main()
{
    int a[SIZE];
    int i;

    printf("\nEnter %d values\n", SIZE);

    // Enter values into the array using pointer notation
    for(i = 0; i < SIZE; i++)
    {
        scanf("%d", & *(a + i));
    } // end for

    // Display the contents of the array using subscript notation and
pointer notation
    for(i = 0; i < SIZE; i++)
    {
        printf("\nElement %d contains %d, same as %d", i, a[i], *(a
+ i));
    } // end for

    return 0;

} // end main()

```

Repl 10.3: <https://replit.com/@michaelTUDublin/103-Pointers-and-Arrays-p3#main.c>