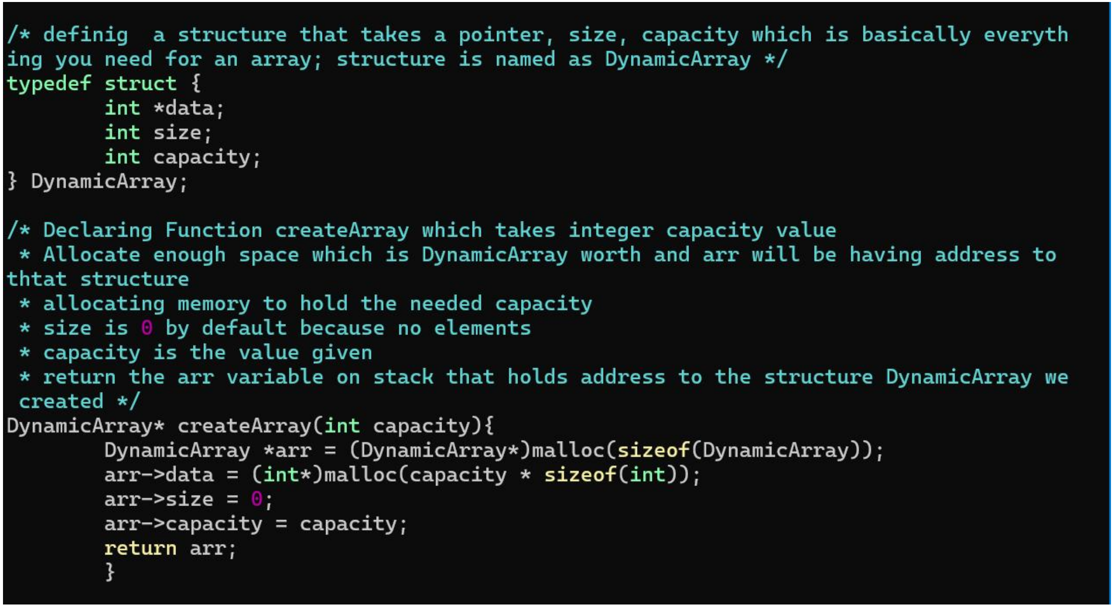
Dynamic Array in C Part 1

Learning Step 1: Create Array

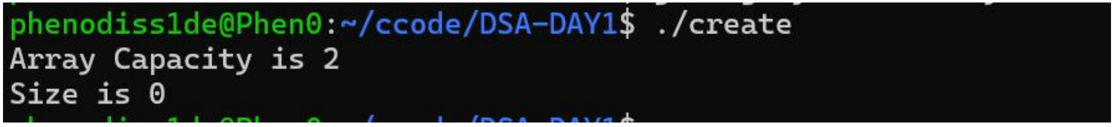
First I started by making createArray function which basically makes an array. Comments can give you a better idea.



```
/* definig  a structure that takes a pointer, size, capacity which is basically everyth
ing you need for an array; structure is named as DynamicArray */
typedef struct {
        int *data;
        int size;
        int capacity;
} DynamicArray;

/* Declaring Function createArray which takes integer capacity value
 * Allocate enough space which is DynamicArray worth and arr will be having address to
thtat structure
 * allocating memory to hold the needed capacity
 * size is 0 by default because no elements
 * capacity is the value given
 * return the arr variable on stack that holds address to the structure DynamicArray we
 created */
DynamicArray* createArray(int capacity){
        DynamicArray *arr = (DynamicArray*)malloc(sizeof(DynamicArray));
        arr->data = (int*)malloc(capacity * sizeof(int));
        arr->size = 0;
        arr->capacity = capacity;
        return arr;
        }
```

compile, debug and then execute.

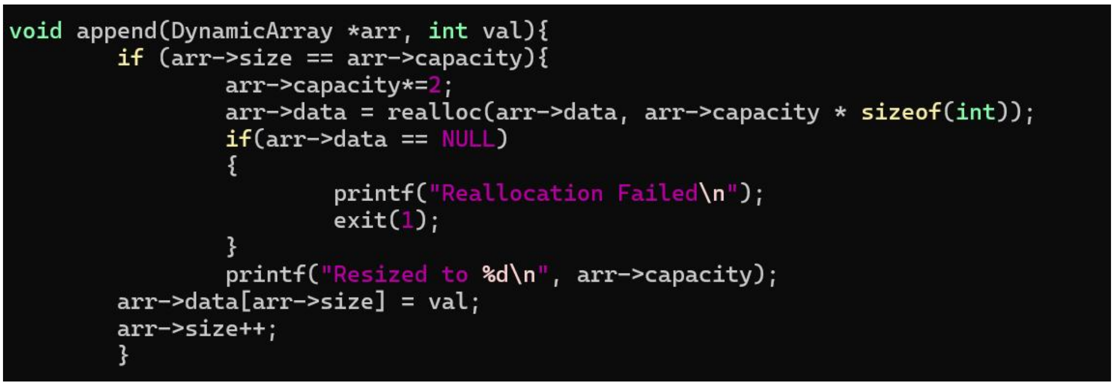


```
phenodiss1de@Phen0:~/ccode/DSA-DAY1$ ./create
Array Capacity is 2
Size is 0
```

Learning Step 2: Append elements in array

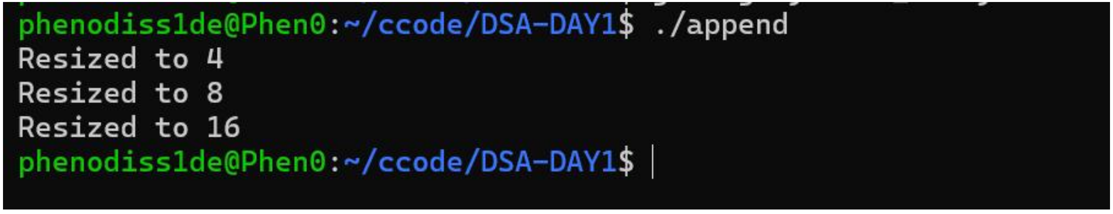
Before moving on we need to understand what is realloc function, think of it like reallocating memory by appening another block to your current chunk of memory. You should always keep track of pointer assignemnt because if memory you add is too big realloc will create new memory not in continuition but at a whole different location.

We start by declaring append function which takes the array and value as input. It checks for the capacity and size comparison, let's say size is 2 and capacity value is 2 it will update by doubling the current capacity value and then reallocate current pointer with new capacity. Then it checks if the pointer is NULL to avoid memory issues. Prints new capacity and then appends value to the correct index and increases size value by 1 (because although it doubles capacity it is to fit 1 integer which is 4 bytes.



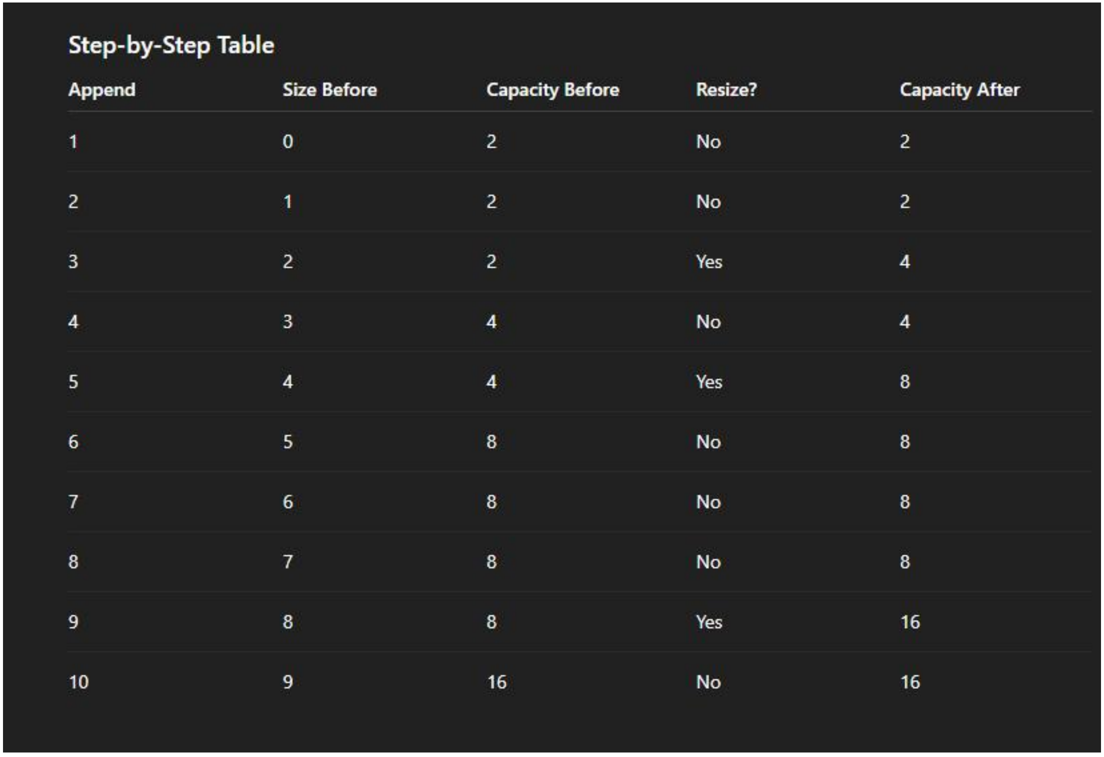
```
void append(DynamicArray *arr, int val){
        if (arr->size == arr->capacity){
                arr->capacity*=2;
                arr->data = realloc(arr->data, arr->capacity * sizeof(int));
                if(arr->data == NULL)
                {
                        printf("Reallocation Failed\n");
                        exit(1);
                }
                printf("Resized to %d\n", arr->capacity);
        arr->data[arr->size] = val;
        arr->size++;
        }
```

On executing we see the resizes at every call of the loop



```
phenodiss1de@Phen0:~/ccode/DSA-DAY1$ ./append
Resized to 4
Resized to 8
Resized to 16
phenodiss1de@Phen0:~/ccode/DSA-DAY1$
```

I got the execution backend from GPT



Step-by-Step Table

Append	Size Before	Capacity Before	Resize?	Capacity After
1	0	2	No	2
2	1	2	No	2
3	2	2	Yes	4
4	3	4	No	4
5	4	4	Yes	8
6	5	8	No	8
7	6	8	No	8
8	7	8	No	8
9	8	8	Yes	16
10	9	16	No	16