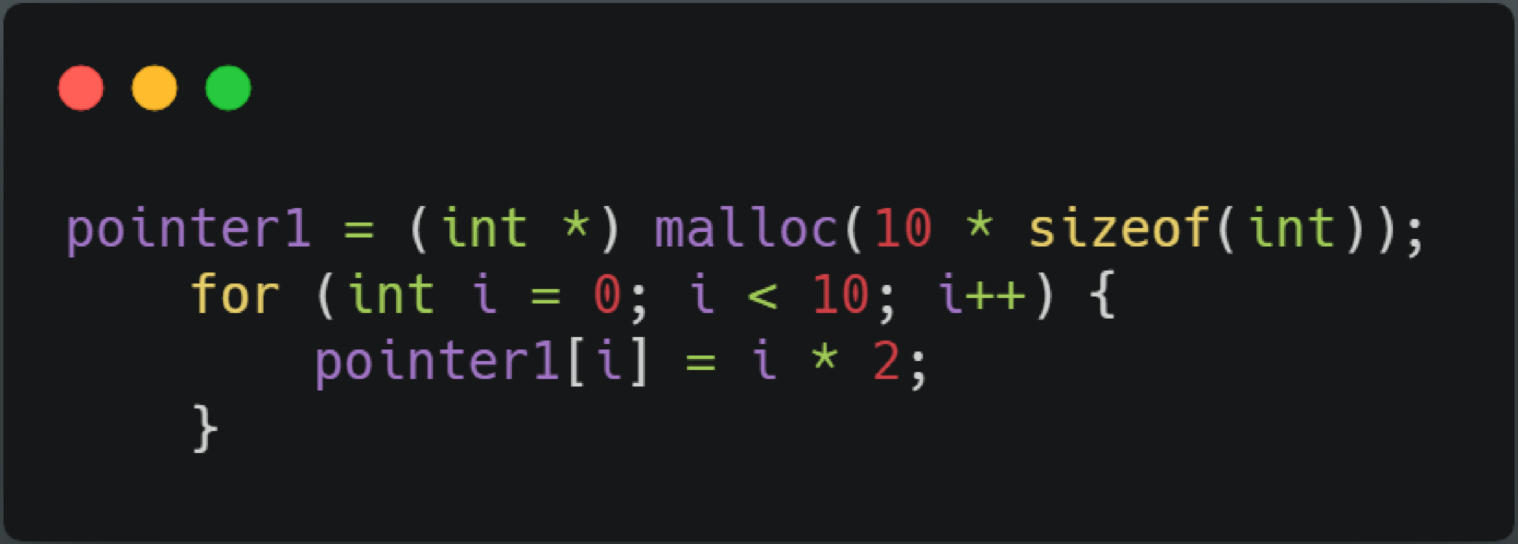
DEMO III

ZOLTÁN NÉMETH

MERGING ARRAYS



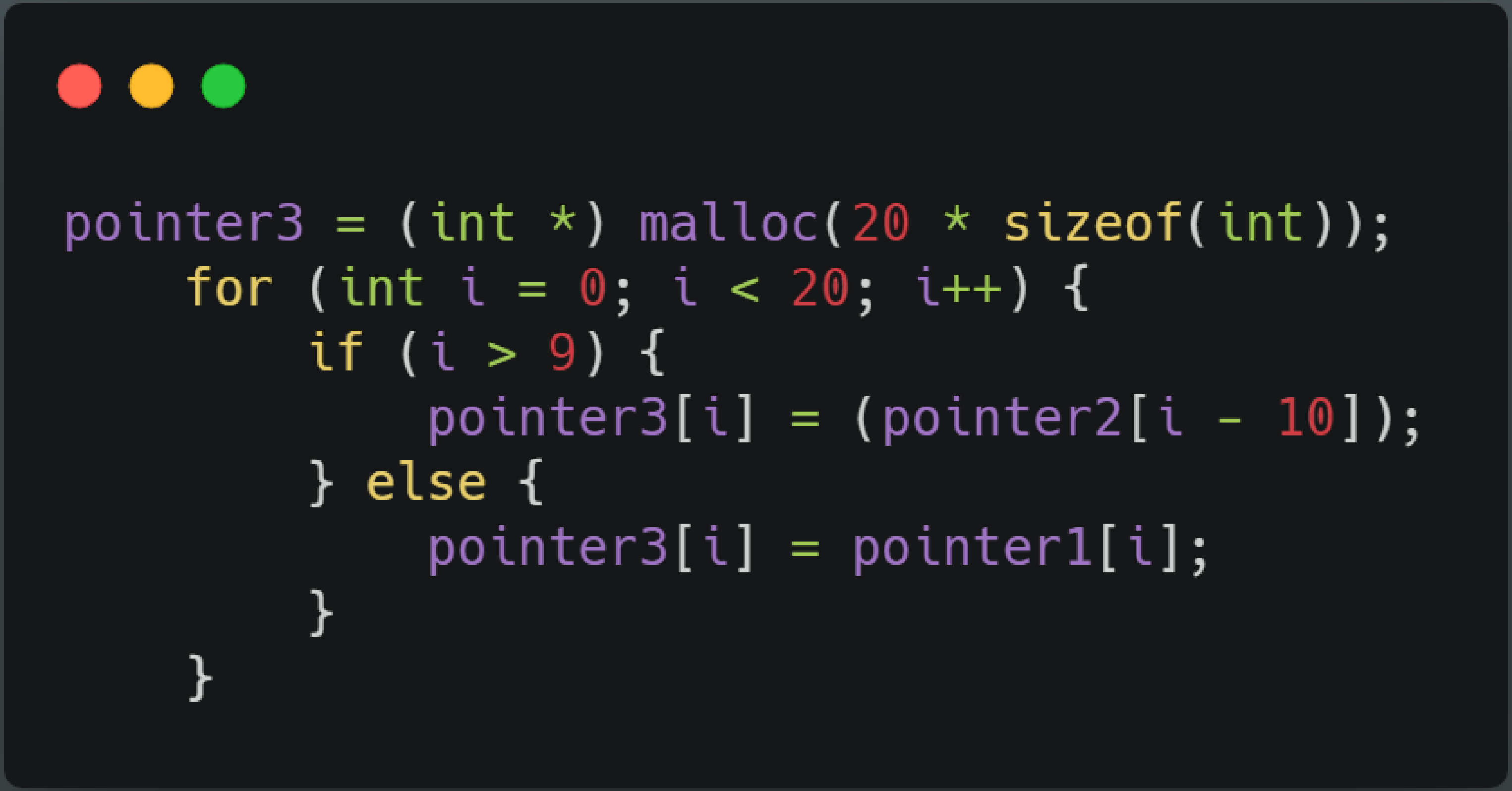
```
pointer1 = (int *) malloc(10 * sizeof(int));
    for (int i = 0; i < 10; i++) {
        pointer1[i] = i * 2;
    }
```

MERGED ARRAY

1. array:
0 2 4 6 8 10 12 14 16 18

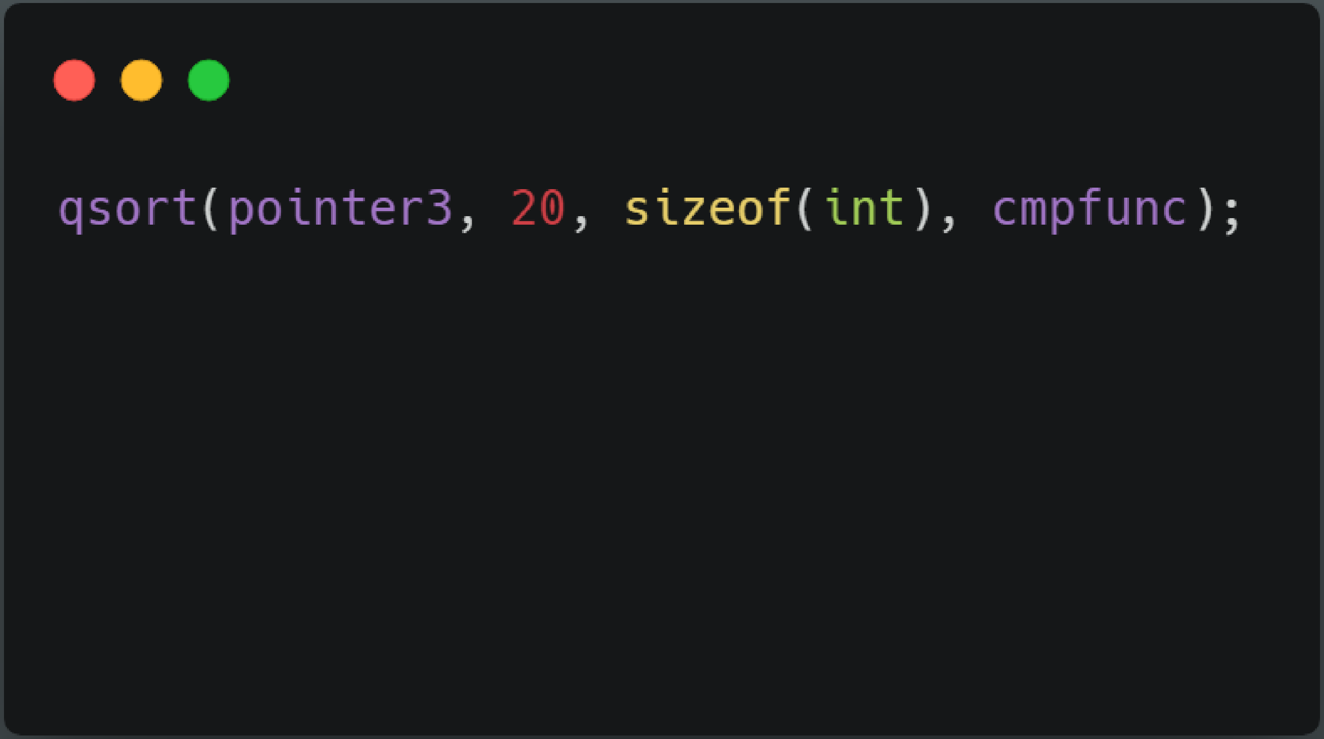
2. array:
1 3 5 7 9 11 13 15 17 19

Merged array:
0 2 4 6 8 10 12 14 16 18 1 3 5 7 9 11 13 15 17 19



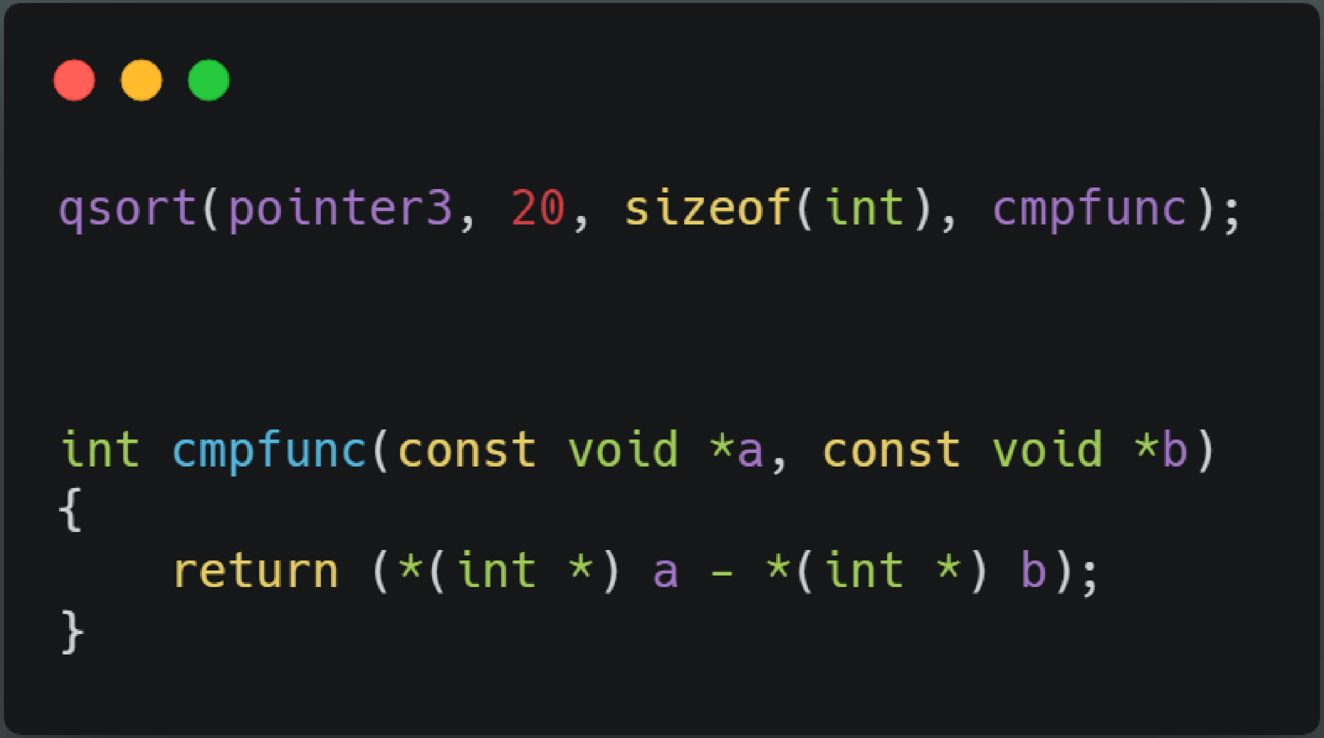
```
pointer3 = (int *) malloc(20 * sizeof(int));
    for (int i = 0; i < 20; i++) {
        if (i > 9) {
            pointer3[i] = (pointer2[i - 10]);
        } else {
            pointer3[i] = pointer1[i];
        }
    }
```

QSORT() FUNCTION



```c
qsort(pointer3, 20, sizeof(int), cmpfunc);
```

QSORT() FUNCTION

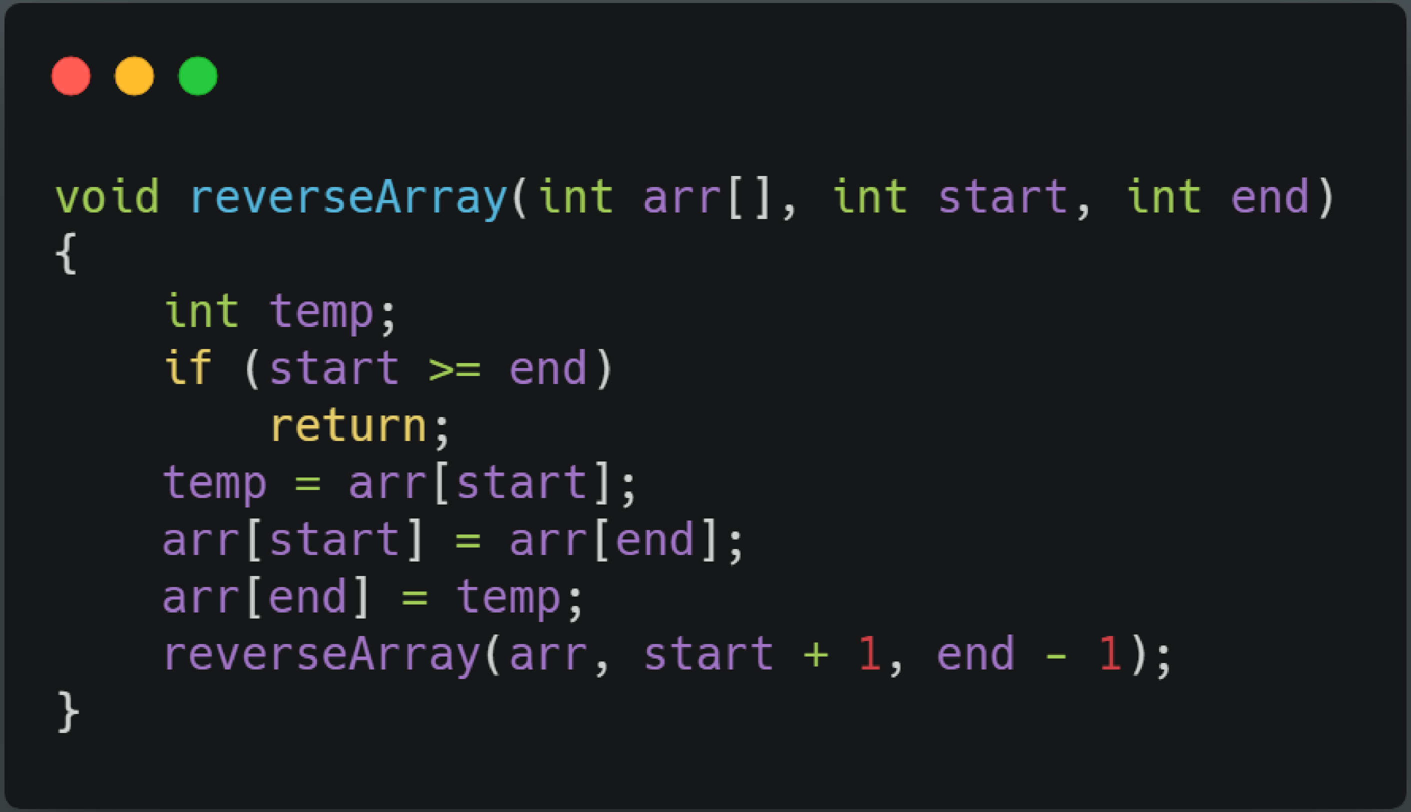


```
qsort(pointer3, 20, sizeof(int), cmpfunc);



int cmpfunc(const void *a, const void *b)
{
    return (*(int *) a - *(int *) b);
}
```

REVERSE ARRAY



```
void reverseArray(int arr[], int start, int end)
{
    int temp;
    if (start >= end)
        return;
    temp = arr[start];
    arr[start] = arr[end];
    arr[end] = temp;
    reverseArray(arr, start + 1, end - 1);
}
```