int upper\_bound (int \*arr, int size, int val)

{

int left = 0, right = size;

while (left < right)

{

int mid = (left + right) / 2;

if (arr[mid] <= val)

left = mid + 1;

else

right = mid;

}

return left;

}

int compare (const void \*a, const void \*b)

{

return \*((int \*)a) - \*((int \*)b);

}

int\* counts(int nums\_count, int\* nums, int maxes\_count, int\* maxes, int\* result\_count) {

// sort nums array

qsort(nums, nums\_count, sizeof(int), compare);

int \*result = malloc(maxes\_count \* sizeof(int));

if (!result)

return NULL;

int index = 0;

for (int i = 0; i < maxes\_count; i++)

{

result[index++] = upper\_bound(nums, nums\_count, maxes[i]);

}

\*result\_count = index;

return result;

}