

```

/**
 * Return an array of arrays of size *returnSize.
 * The sizes of the arrays are returned as *returnColumnSizes array.
 * Note: Both returned array and *columnSizes array must be malloced, assume caller calls free().
 */

void _permute(int* nums, int numsSize, int* returnSize, int** returnColumnSizes, int** result, int* stack, int stack_ptr)
{
    int index;
    int tmp;

    for(index = 0; index < numsSize; index++)
    {
        if(nums[index] != INT_MIN)
        {
            stack[stack_ptr+1] = nums[index];

            if( (stack_ptr+1) == (numsSize-1) )
            {
                result[*returnSize] = (int*)malloc(sizeof(int)*numsSize);
                memcpy(result[*returnSize], stack, sizeof(int)*numsSize);
                (*returnColumnSizes)[*returnSize] = numsSize;
                (*returnSize)++;
            }else
            {
                tmp = nums[index];
                nums[index] = INT_MIN;

                _permute(nums, numsSize, returnSize, returnColumnSizes, result, stack, stack_ptr+1);
                nums[index] = tmp;
            }
        }
    }
}

int** permute(int* nums, int numsSize, int* returnSize, int** returnColumnSizes) {
    int** result;
    int index;
    int alloc_length;
    int stack[6];
    alloc_length = 1;
    *returnSize = 0;
    for(index = numsSize; index > 0; index--)
    {
        alloc_length *= index;
    }

    result = (int**)malloc(sizeof(int*)*alloc_length);
    *returnColumnSizes = (int*)malloc(sizeof(int)*alloc_length);
    _permute(nums, numsSize, returnSize, returnColumnSizes, result, stack, -1);

    return result;
}

```