

## Atestarea nr. 2 la disciplina „Algoritmi, Structuri de Date și Complexitate” 23.04.2021, ora 9:45 – 11:15

### Входные данные

Создайте 10-ти элементный одномерный массив **MyArray** состоящий из элементов типа **char**, заполненный латинскими буквами вашего имени. Если ваше имя состоит из менее 10-ти букв, используйте отчество и фамилию. Также, создайте массив **MySortedArray** который совпадает с отсортированным в алфавитном порядке массивом **MyArray**.

### Примеры

1. Если ваше имя «Александрина» («Alexandrina»), тогда:

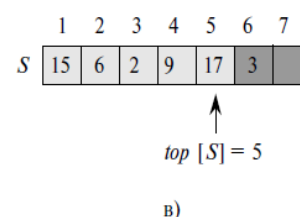
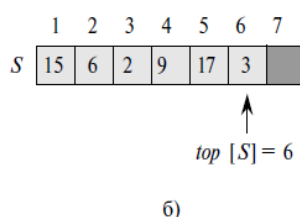
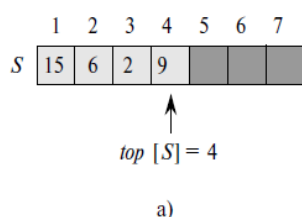
**MyArray** = { a, l, e, x, a, n, d, r, i, n }  
**MySortedArray** = { a, a, d, e, i, l, n, n, r, x }

2. Если ваша фамилия «Иван» («Ivan»), то добавьте ваше отчество, например «Петрович» («Petrovici»). В случае необходимости добавьте и начало фамилии:

**MyArray** = { i, v, a, n, p, e, t, r, o, v }  
**MySortedArray** = { a, e, i, n, o, p, r, t, v, v }

### Задания

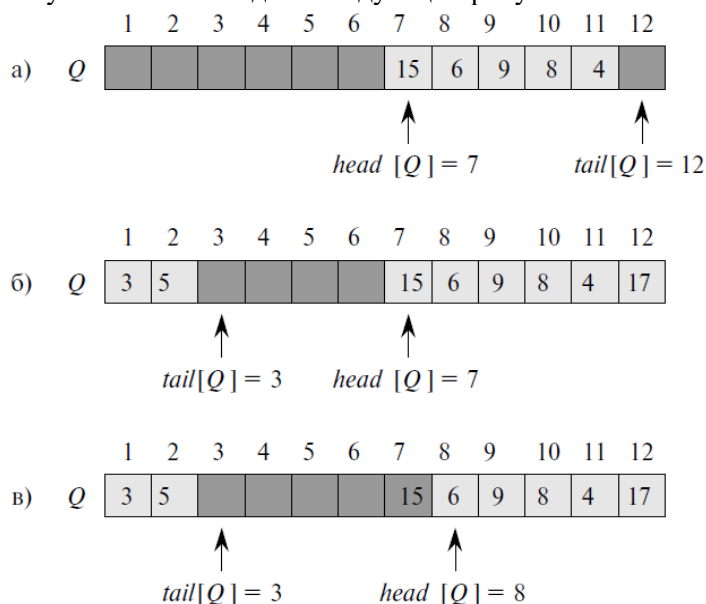
1. Примените метод Фибоначчи для поиска элемента 'a' в вашем персональном массиве **MySortedArray**. Перечислите все элементы из массива **MySortedArray** с которыми был сопоставлен искомый элемент в процессе поиска (в порядке сопоставления), независимо от результата поиска (был ли найден элемент, или нет). **(15 б.)**
2. Примените Сортировку слиянием (MergeSort) над вашим персональным массивом **MyArray**, сортируя элементы в алфавитном порядке. Укажите детально каждое прохождение массива данным методом. Для каждого прохода, укажите количество сравнений и перестановок, которые были произведены для его получения (для исходного массива ничего не указывается). **(15 б.)**
3. Из элементов персонального массива **MyArray** постройте бинарное дерево поиска. Проиллюстрируйте выполнение трёх типов обходов: «лкп», «клп», «лпк». **(15 б.)**
4. Используя в качестве модели следующий рисунок:



проиллюстрируйте результат воздействия на изначально пустой стек **S**, хранящийся в

массиве  $S[1..6]$ , операций **PUSH(S, 4)**, **PUSH(S, 1)**, **PUSH(S, 3)**, **POP(S)**, **PUSH(S, 8)** и **POP(S)**. (10 6.)

5. Используя в качестве модели следующий рисунок:



проиллюстрируйте результат воздействия на изначально пустую очередь  $Q$ , хранящуюся в массиве  $Q[1..6]$ , операций **ENQUEUE(Q, 4)**, **ENQUEUE(Q, 1)**, **ENQUEUE(Q, 3)**, **DEQUEUE(Q)**, **ENQUEUE(Q, 8)** и **DEQUEUE(Q)**. (10 6.)

6. Покажите, как реализовать очередь с помощью двух стеков. Проанализируйте время работы операций, которые выполняются с ее элементами. (10 6.)
7. Напишите код программы в которой создается линейный динамический двусвязный список, элементов типа **char**. Список будет храниться **одним единственным указателем на любой его элемент**. Левый край списка будет считаться его началом, правый край – его концом. Напишите следующие методы: а) Позиционирование текущего указателя на первый элемент списка, б) Вывод элементов списка слева-направо, начиная с **текущего** элемента, в) Добавление одного нового элемента в начало списка.

В методе **main**, создайте пустой список. Затем добавьте первые четыре элемента вашего персонального массива **MyArray**, используя метод (в). Затем выведите содержимое списка, используя методы (а) и (б).

Укажите что будет выведено на консоль.

(25 6.)

## Barem de notare

Nota	1	2	3	4	5	6	7	8	9	10
Punctaj acumulat	E	0--10	11--20	21--29	30--45	46--60	61--75	76--85	86--95	96--100

Examinator: Valeriu Ungureanu, dr., conf. univ.