

Laborator 5. Stive si cozi

Subiecte Marti 16-18

Stiva si coada sunt tipuri speciale de liste, mai usor de scris – in stiva se poate adauga/ sterge doar la inceput, in coada se adauga la final, se sterge de la inceput.

1. Pornind de la urmatoarele definitii:

```
typedef struct Stack {
    struct node *top;
    int len;
} Stack;

typedef struct Queue {
    struct node *head, *tail;
    int len;
} Queue;

typedef struct node {
    int data;
    struct node *next;
} node;
```

Sa se creeze functii pentru:

- initierea unei stive / cozi

```
void init(Stack ** s);
void init(Queue **q);
```

- dezallocarea memoriei si stergerea stivei/ cozii:

```
void free(Stack **s);
void free(Queue **s);
```

- adaugare element in stiva/ coada (intoarce 1 daca s-a putut adauga, altfel 0):

```
int add_stack (Stack *s, int val);
int add_queue (Queue *q, int val);
```

- stergere element din stiva/ coada (intoarce elementul sters):

```
int pop_stack(Stack *s);
int pop_queue(Queue *q);
```

- afisare stiva/ coada:

```
void print(Stack *s);
void print(Queue *q);
```

- verificare daca stiva/ coada e vida:

```
int is_empty(Stack *s);  
int is_empty(Queue *q);
```

2. Sortati un vector cu ajutorul algoritmului RadixSort, folosind 10 cozi – o coada pentru fiecare cifra. La fiecare pas p, se adauga in coada k elementele care au cifra p (pornind de la dreapta la stanga) egala cu k, apoi se parcurg cozile si se adauga in ordine elementele in vector. La final, vectorul devine sortat. Mai multe explicatii si exemple gasiti aici [1].

3. Dandu-se o expresie cu adunari, scaderi, inmultiri, impartiri si paranteze, aflati rezultatul expresiei folosind doua stive. Descrierea algoritmului se gaseste aici [2].

1 2 3 4 5 6 7 8 9 10

3->4->5->6->7->8->9->10->11->NULL

[1] https://en.wikipedia.org/wiki/Radix_sort

[2] <https://www.geeksforgeeks.org/expression-evaluation/>