# Лабораторная работа №1

## Подготовка к работе

1. Реализация дека.

Определены следующие процедуры и функции:

* Функция *Empty*(*D*) возвращает **true**, если дек пуст, **false** в противном случае, сложность *O*(1);
* Функция *PushBack*(*D*, *x*) помещает элемент *x* в конец дека *D*, сложность *O*(1);
* Функция *PushFront*(*D*, *x*) помещает элемент *x* в начало дека *D*, сложность *O*(1);
* Функция *PopBack*(*D*) удаляет элемент с конца дека *D* и возвращает его значение (не определена для пустого дека), сложность *O*(1);
* Функция *PopFront*(*D*) удаляет элемент из начала дека *D* и возвращает его значение (не определена для пустого дека), сложность *O*(1);
  1. Массив.

Массив из *m* компонентов *D*0, *D*1, *D*2, …, *Dm*–1, свёрнутый в кольцо в соответствии с операцией **mod**. Считается, что *D*0 следует за *Dm*–1. Переменная *f* – указатель позиции в массиве, расположенной непосредственно перед началом дека, переменная *r* – указатель позиции в массиве, расположенной после конца дека, т. е. дек состоит из элементов *Df*+1, *Df*+2, …, *Dr-*1.

|  |
| --- |
| *Empty*(*D*) |
|  |

|  |  |
| --- | --- |
| *PushBack*(*D*, *x*) | *PushFront*(*D*, *x*) |
|  |  |

|  |  |
| --- | --- |
| *PopBack*(*D*) | *PopFront*(*D*) |
|  |  |

* 1. Связный список.

Используется связанный список с заголовком, узел которого состоит из поля *info*, содержащего элемент дека, и полей связи *prev* и *next* для указания элемента, идущего до и после данного соответственно. При таком представлении дека поле *info* заголовка не используется, поле *next* заголовка указывает на первый элемент очереди, поле *prev* заголовка указывает на последний элемент очереди, переменная *header* обозначает заголовок дека. Пустой дек состоит из одного узла-заголовка и значения его полей *next* и *prev* равно Λ.

|  |
| --- |
| *Empty*(*D*) |
|  |

|  |  |
| --- | --- |
| *PushBack*(*D*, *x*) | *PushFront*(*D*, *x*) |
|  |  |

|  |  |
| --- | --- |
| *PopBack*(*D*) | *PopFront*(*D*) |
|  |  |

1. Два стека в одном массиве.

Определены следующие процедуры и функции:

* Функция *Empty1*(*S*) возвращает **true**, если первый стек пуст, **false** в противном случае, сложность *O*(1);
* Функция *Empty2*(*S*) возвращает **true**, если второй стек пуст, **false** в противном случае, сложность *O*(1);
* Функция *Push1*(*S*, *x*) помещает элемент *x* в конец первого стека, сложность *O*(1);
* Функция *Push2*(*S*, *x*) помещает элемент *x* в конец второго стека, сложность *O*(1);
* Функция *Pop1*(*S*) удаляет элемент с конца первого стека и возвращает его значение (не определена для пустого стека), сложность *O*(1);
* Функция *Pop2*(*S*) удаляет элемент с конца второго стека и возвращает его значение (не определена для пустого стека), сложность *O*(1);

Массив из *m* компонентов *S*0, *S*1, *S*2, …, *Sm*–1. Переменная *p*1 – указатель позиции в массиве, расположенной после конца первого стека, переменная *p*2 – указатель позиции в массиве, расположенной после конца второго стека. Первый стек состоит из элементов *S*0, *S*1, …, *Sp1*–1, а второй – *Sm*–1, *Sm*–2, …, *Sp2*+1.

|  |  |
| --- | --- |
| *Empty1*(*S*) | *Empty2*(*S*) |
|  |  |

|  |  |
| --- | --- |
| *Push1*(*S*, *x*) | *Push2*(*S*, *x*) |
|  |  |

|  |  |
| --- | --- |
| *Pop1*(*S*) | *Pop2*(*S*) |
|  |  |

1. Очередь на базе двух стеков.

Определены следующие процедуры и функции:

* Функция *Empty*(*Q*) возвращает **true**, если очередь пуста, **false** в противном случае, сложность *O*(1);
* Функция *Push*(*Q*, *x*) помещает элемент *x* в конец очереди *Q*, сложность *O*(1);
* Функция *Pop*(*Q*) удаляет элемент из начала очереди *Q* и возвращает его значение (не определена для пустой очереди), сложность *O*(*N*);

Стек *S*1 содержит данные, а стек *S*2 используется для операции *Pop*(*Q*). Для стеков *S*1 и *S*2 также определены функции *Empty*(*S*), *Push*(*S*, *x*) и *Pop*(*S*).

|  |  |  |
| --- | --- | --- |
| *Empty*(*Q*) | *Push*(*Q*, *x*) | *Pop*(*Q*) |
|  |  |  |

1. Стек на базе двух очередей.

Определены следующие процедуры и функции:

* Функция *Empty*(*S*) возвращает **true**, если стек пуст, **false** в противном случае, сложность *O*(1);
* Функция *Push*(*S*, *x*) помещает элемент *x* в конец стека *S*, сложность *O*(1);
* Функция *Pop*(*S*) удаляет элемент с конца стека *S* и возвращает его значение (не определена для пустого стека), сложность *O*(*N*);

Очередь *Q*1 содержит данные, а стек *Q*2 используется для операции *Pop*(*S*). Для очередей *Q*1 и *Q*2 также определены функции *Empty*(*S*), *Push*(*S*, *x*) и *Pop*(*S*).

|  |  |  |
| --- | --- | --- |
| *Empty*(*S*) | *Push*(*S*, *x*) | *Pop*(*S*) |
|  |  |  |

1. Алгоритмы решения задач.

Блок 1.

|  |
| --- |
| Префикс → Постфикс |
|  |

Блок 2.

|  |
| --- |
| Цифры в конец |
|  |

Выводы: