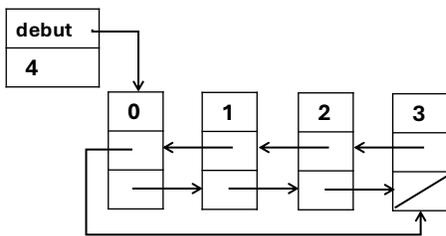


UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 339

Série d'exercices - Thème #8 : Conteneurs linéaires non contigus

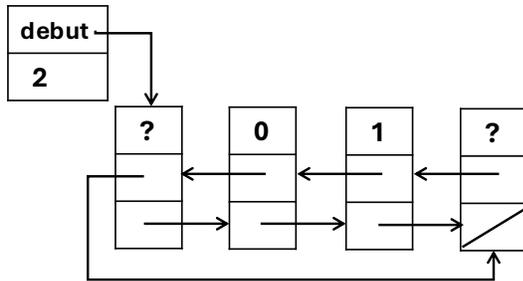
Exercice 1 :



Écrire le code des fonctions `push_back` et `pop_back` pour cette représentation de la `list` (liste), sans cellules fictives de début et fin.

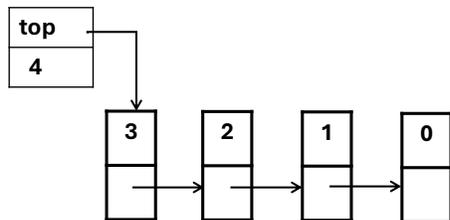
```
template <typename TYPE>
class cell{
private:
    TYPE val;
    cell* prec, suiv;
public:
    cell(TYPE&);
}
template <typename TYPE>
class list{
private:
    cell* debut;
    size_t dim;
public:
    ...
    void push_back(TYPE val);
    void pop_back();}
```

Exercice 2 :



Écrire le code des fonctions **push_back** et **pop_back** pour cette représentation de la **list** (liste), avec des cellules fictives de début et fin.

Exercice 3 :



Écrire le code des fonctions **push**, **pop** et **top** pour cette représentation de la **stack** (pile).

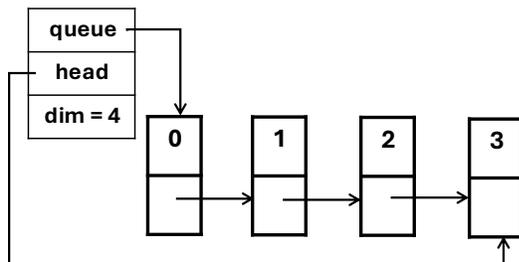
```
template <typename TYPE>
class cell{
private:
    TYPE val;
    cell* suiv;
public:
    cell(TYPE&);
}
template <typename TYPE>
class stack{
private:
    cell* top;
    size_t dim;
public:
```

```

...
void push(TYPE val);
void pop();
TYPE top();
}

```

Exercice 4 :



Écrire le code des fonctions **enqueue** et **dequeue** pour cette représentation de la **queue** (file). Note : on insère à la tête (head), et on retire à la queue.

```

template <typename TYPE>
class cell{
private:
    TYPE val;
    cell* suiv;
public:
    cell(TYPE&);
}
template <typename TYPE>
class queue{
private:
    cell* queue;
    cell* head;
    size_t dim;
public:
    ...
    void enqueue(TYPE val);
    void dequeue();
}

```