

Exercices: Introduction aux structures de données

Algorithmics & data structures – ESTIAM 2 eme année – Octobre 2024

Liste chaînée

1 – Définir les structures permettant de représenter une liste chaînée

```
typedef struct n{
    int data;
    struct n* next;
} Node;
```

2 – Écrire une fonction qui prend en paramètre la tête d'une liste chaînée et qui inverse cette liste

Pile

3 – Implémenter une pile en utilisant un tableau

Définir la structure d'une pile et implémentez les fonctions push et pop.

```
void push(Stack s, int value);

int pop(Stack s)
```

File

3 – implémenter une file en utilisant une liste chaînée.

Définir une structure qui représente une file stockant des entiers

```
typedef struct{
    ???
} Queue;
```

Réfléchissez aux données dont vous avez besoin.

Écrire une fonction void push(Queue q, int value) qui permet d'ajouter une valeur à une File

Écrire une fonction int get(Queue q) qui recupère une valeur d'une file. La fonction doit retourner NULL si la file est vide

Écrire une fonction `int is_empty(Queue q)` qui retourne 0 si la file est vide et 1 sinon

Algorithme

4 – implémenter une fonction qui inverse une string en utilisant une stack

```
void reverse_string(char str[])
```

5 – implémenter une fonction qui inverse l'ordre des mots d'une phrase en utilisant une queue

Note : Vous pouvez utiliser la fonction `strtok` pour découper les mots

```
void reverse_words(char str[]);
```