

Travaux Pratiques : Implémentation et utilisation d'une pile (Stack) en Langage C

DUT : IDSD

2024-2025

I. Objectifs du TP

- Comprendre et manipuler la structure de données pile
- Comparer deux approches d'implémentation : tableau vs liste chaînée
- Analyser les avantages et contraintes de chaque méthode

II. Énoncé du TP

Dans ce travail pratique, vous devez :

1. Implémenter une pile à l'aide d'un tableau (taille fixe ou dynamique)
2. Implémenter une pile à l'aide d'une liste chaînée
3. Développer un programme de test pour valider les opérations suivantes :
 - `push`, `pop`, `top`, `isEmpty`
4. Rédiger un court rapport (1 page) comparant les deux implémentations en termes de :
 - Performance
 - Complexité mémoire
 - Facilité de mise en œuvre

III. Questions et prolongements

- Quelle implémentation offre la meilleure gestion mémoire ?
- Comment adapter la version tableau pour qu'elle soit extensible ?
- Proposer une version générique (`void*`) permettant de stocker différents types de données
- Étendre le TP pour écrire un parseur d'expression postfixée utilisant votre pile

IV. En-têtes de fonctions et explications

1. Pile via un tableau

```

1 // Initialise la pile (top = -1)
2 void initStack(Stack *s);
3
4 // Teste si la pile est vide (top == -1)
5 int isEmpty(Stack *s);
6
7 // Teste si la pile est pleine (top == MAX_STACK-1)
8 int isFull(Stack *s);
9
10 // Empile l'element x (verifie si pleine)
11 void push(Stack *s, int x);
12
13 // Depile et retourne l'element au sommet
14 int pop(Stack *s);
15
16 // Retourne sans depiler l'element au sommet
17 int peek(Stack *s);

```

2. Pile via une liste chaînée

```

1 // Initialise la pile (s->top = NULL)
2 void initStackList(StackList *s);
3
4 // Teste si la pile est vide (s->top == NULL)
5 int isEmptyList(StackList *s);
6
7 // Empile l'element x en tete de liste
8 void pushList(StackList *s, int x);
9
10 // Depile et retourne l'element en tete
11 int popList(StackList *s);
12
13 // Retourne sans depiler l'element en tete
14 int peekList(StackList *s);
15
16 // Libere tous les noeuds de la pile
17 void freeStack(StackList *s);

```