

# Institut Supérieur d'Informatique et de Mathématiques

## DS ASD2

Classe : L1 Info

Enseignant : Taoufik Sakka Rouis

Nom et Prénom:.....

CIN : .....

A.U. : 2024/2025

Durée : 1H00

Nombre des Pages : 4

Documents Autorisés : Non

### Exercice 1 : (3 points)

Enrichir l'OA file en ajoutant la méthode Multi\_Def (unsigned k) qui permet de défiler au maximum k éléments de la file.

```
struct Cellule {  
    int cle ;  
    struct Cellule * suivant ;  
};  
struct file {  
    struct Cellule * tete ;  
    struct Cellule * queue ;  
};  
static struct file f;
```

Ne rien écrire ici

### Exercice 2 : (2+3+12 Points)

La notation polonaise inverse (NPI) permet d'écrire de façon non ambiguë les formules arithmétiques sans utiliser de parenthèses.

Par exemple l'expression «  $1 + 6 * 5$  » s'écrit «  $1\ 6\ 5\ * +$  », l'expression «  $(1 + 6) * 5$  » s'écrit «  $1\ 6\ +\ * 5$  » et enfin l'expression «  $(2 * (4 + 6))/5$  » s'écrit «  $2\ 4\ 6\ +\ * 5\ /\$  ».

**Question 1 :** Ecrire en NPI les expressions suivantes :

$(1 + 6) * (5 + 4 * 3)$  → .....  
 $((1 + 2) * 3 + 4) * 5$  → .....  
 $(1 + 6) - (5 + 9 * 4)$  → .....  
 $9 * 5 / (4 + 3)$  → .....

**Question 2 :** Evaluer les expressions suivantes écrites en NPI :

$5\ 10\ + =$ .....	$2\ 10\ 4\ +\ - =$ .....
$10\ 4\ +\ 2\ - =$ .....	$4\ 3\ +\ 3\ 2\ +\ * =$ .....
$51\ 8\ + =$ .....	$11\ -2\ * =$ .....

### Question 3 :

On cherche à écrire une fonction de prototype `int evaluerExpression (char *exp)`, permettant d'évaluer une expression donnée en notation polonaise inverse (NPI). Cette expression est **supposée valide** : correcte en notation NPI et ses opérateurs ainsi que ses opérandes sont séparés par un espace. Par exemple, l'expression représentée par la chaîne "**10 4 + 2 -**" devra être évaluée correctement à **12**.

### Remarque:

- Vous pouvez importer et utiliser la SD appropriée (pile OA, pile TDA, file OA, etc.).
- Vous pouvez utiliser la fonction `isspace(<c>)` de la bibliothèque `<ctype.h>`. Cette fonction permet de vérifier si le caractère `<c>` est un espace blanc.
- Vous pouvez utiliser la fonction `atoi(<s>)` de la bibliothèque `<stdlib.h>`. Cette fonction retourne la valeur numérique représentée par la chaîne `<s>` sous forme d'entier.
- Vous pouvez utiliser la fonction de prototype `char* substring(char *source, int start, int end)` pour extraire une sous-chaîne à partir de deux indices (start et end) :

### exemple :

```
char *text = "Bonjour tout le monde!";  
char *result = substring (text, 8, 11);  
printf ("Sous-chaîne : %s\n", result); /* affiche Sous-chaîne : tout */
```



