Examen 2022-2023 - CYBER1 (2h00)

Algo et Structure de Données 1

NOM:	PRÉNOM :

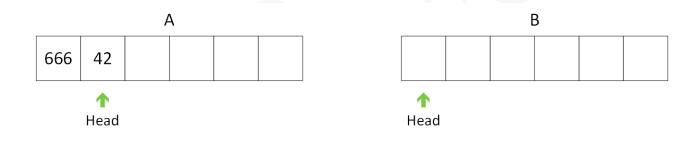
Vous devez respecter les consignes suivantes, sous peine de 0:

- Lisez le sujet en entier avec attention
- Répondez sur le sujet
- Ne détachez pas les agrafes du sujet
- Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- Vous devez écrire dans le langage algorithmique classique ou en C (donc pas de Python ou autre)
- Ne trichez pas

1 Questions (5 points)

1.1 (1 points) Écrivez l'état des deux piles après avoir effectué ces opérations dans cet ordre (n'oubliez pas le(s) pointeur(s) de tête (et de queue)) :

empiler 1337, empiler 2048, dépiler, dépiler, empiler 1024, dépiler, empiler 1664



666	42	1664					1664					
-----	----	------	--	--	--	--	------	--	--	--	--	--

 $1.2 \quad (0,5 \text{ point})$ Quel élément sortira lors du prochain « pop » sur chaque pile?

A: 1664 B: 1664

1.3 (0,5 point) Quel élément sortira en dernier de chaque pile?

A: 666 B: 1664

1.4 (3 points) En admettant que l'on dispose d'une pile et que l'on insère les données « 1 2 3 4 5 6 » dans cet ordre exclusivement, décrivez les scénarios permettant d'obtenir les sorties suivantes :

```
exemple: pour « A B C » en entrée, on peut obtenir « B C A » en sortie en faisant : « push A », « push B », « pop », « push C », « pop », « pop »

3, 2, 1, 4, 5, 6

push 1, push 2, push 3, pop, pop, pop, push 4, pop, push 5, pop, push 6, pop

1, 4, 3, 5, 2, 6

push 1, pop, push 2, push 3, push 4, pop, pop, push 5, pop, pop, push 6, pop

2, 4, 3, 5, 6, 1

push 1, push 2, pop, push 3, push 4, pop, pop, push 5, pop, push 6, pop, pop
```

- 2 Algorithmes (15 points)
- 2.1 (1,5 points) Écrivez une structure de données « my_stack_t » pouvant servir de pile et stockant les éléments dans un tableau

```
struct my_stack_t
  int[] tab
  int max_len
  int nb_elt
  (int head)
fin struct
```

2.2 (1,5 points) Écrivez une structure de données « my_queue_p » pouvant servir de file et stockant les éléments dans une liste chaînée avec pointeurs

2.3	(3 points) Écrivez une fonction « push » pouvant servir à empiler un élément
	dans votre précédente structure « my $stack$ t »

Vérifier si :

- pointeur NULL est géré en paramètre,
- pile **pleine** est gérée en paramètre,
- et le cas normal (malloc, et suppression en tête avec réorganisation des éléments suivant).
- 2.4 (3 points) Écrivez une fonction « pop » pouvant servir à dépiler un élément dans votre précédente structure « my_stack_t »

Vérifier si :

- pointeur NULL est géré en paramètre,
- pile **vide** est gérée en paramètre,
- et le cas normal (free, et suppression en tête avec réorganisation des éléments suivant).
- 2.5 (3 points) Écrivez une fonction « enqueue » pouvant servir à enfiler un élément dans votre précédente structure « my_queue_p »

Vérifier si :

- pointeur NULL est géré en paramètre,
- file **pleine** est gérée en paramètre,
- et le cas normal (**malloc**, et suppression en tête/queue avec réorganisation des éléments suivant).
- 2.6 (3 points) Écrivez une fonction « dequeue » pouvant servir à défiler un élément dans votre précédente structure « my_queue_p »

Vérifier si :

- pointeur NULL est géré en paramètre,
- file **vide** est gérée en paramètre,
- et le cas normal (free, et suppression en tête/queue avec réorganisation des éléments suivant).