

Examen 2021-2022 - CYBER1 (1h30)

Algorithmique II

NOM :

PRÉNOM :

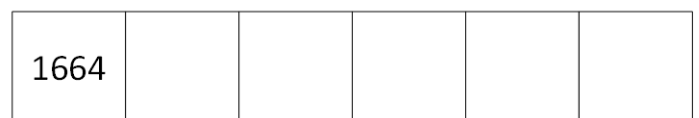
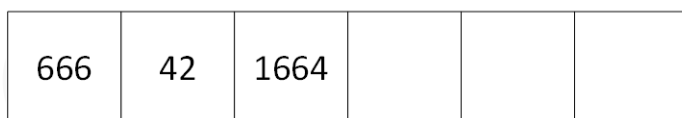
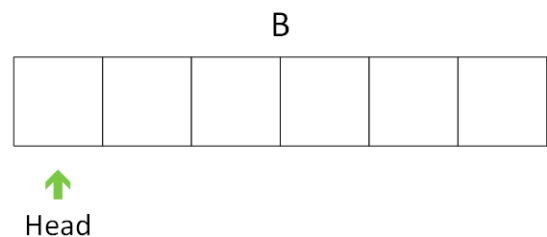
Vous devez respecter les consignes suivantes, sous peine de 0 :

- Répondez sur le sujet
- Ne détachez pas les agrafes du sujet
- Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- 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



1.2 (0,5 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 (3 points) Écrivez une structure de données « *my_stack* » pouvant servir de pile (*la structure ne doit pas être statique*)

[La structure minimaliste en liste chaînée est tolérée, tout comme les structures plus complexes sous forme de conteneur avec des attributs]

```
struct my_stack_1 {    // Structure minimaliste
    int/void* elt,
    my_stack_1 *next
};

struct my_stack_2 {    // Conteneur liste chainee
    int      nb_elt,
    [my_stack_1 *head,]
    my_stack_1 *stack
};

struct my_stack_3 {    // Conteneur tableau
    int      nb_elt,
    int      max_len,
    [int     head,]
    int      *array
};
```

2.2 (6 points) Écrivez une fonction « *push* » pouvant servir à empiler un élément dans votre précédente structure « *my_stack* »

2 cas possibles : tableau OU liste chaînée

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).

2.3 (6 points) Écrivez une fonction « *pop* » pouvant servir à dépiler un élément dans votre précédente structure « *my_stack* »

2 cas possibles : tableau OU liste chaînée

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).