

LIFAPSD - Evaluation de TP #2 – Groupes ABCD – Sujet A

Date : lundi 28 novembre 2022 - Durée : 25 minutes

Vous avez le droit à vos notes de cours et de TD, ainsi qu'à vos TP précédents. Les plagiatés seront sanctionnés par un zéro pour les étudiants concernés.

Enoncé : l'objectif de cet exercice est d'ajouter des fonctionnalités à la classe `TableauDynamique` pour gérer des ensembles de valeurs entières.

1. Créer et ouvrir un fichier dont le nom est basé sur votre nom sous la forme : `nomEtudiant_TP2.cpp` dans un nouveau répertoire.
2. Récupérer sur le site de l'UE les fichiers `TableauDynamique.h`, `TableauDynamique.cpp`, `ElementTD.h` et `ElementTD.cpp` du TP5 et placer les dans ce nouveau répertoire.
3. Dans le fichier `TableauDynamique.h`, écrivez la déclaration de la nouvelle fonction membre suivante :

Fonction `unionTab (tab : Tableau dynamique d'entiers) : lien sur Tableau dynamique d'entiers`
Précondition : l'instance et `tab` sont des tableaux dynamiques contenant chacun des entiers distincts
Résultat : l'adresse d'un nouveau tableau dynamique, alloué sur le tas, qui contient l'union des entiers de l'instance et de `tab`. Attention, le tableau résultat ne contient donc pas de doublon.
Paramètres en mode donnée : `tab`

Exemple : `tab1.unionTab(tab2)` avec `tab1=[-1,2,5]` et `tab2=[5,-2,3,2]` retourne l'adresse d'un nouveau tableau dynamique contenant `[5,-2,3,2,-1]`

4. Dans votre fichier `nomEtudiant_TP2.cpp`, écrivez le code de cette fonction membre.
5. Toujours dans votre fichier `nomEtudiant_TP2.cpp`, écrivez un programme principal permettant de :
 - Créer sur la pile deux tableaux dynamiques nommés `tableau1` et `tableau2`
 - Ajouter quelques valeurs dans ces deux tableaux dynamiques
 - Afficher les deux tableaux dynamiques
 - Faire l'union des deux tableaux
 - Afficher le tableau contenant cette union

Rendu : déposez le fichier `nomEtudiant_TP2.cpp` sur Tomuss dans la case « DepotTP2-ABCD » de LIFAPSD. Prévenez votre enseignant lorsque votre dépôt a été fait. Tout dépassement de l'heure limite sera sanctionné par un zéro.

10/10

LIFAPSD - Evaluation de TD #1 – Groupes ABCD – sujet B

Date : lundi 26 septembre 2022 - Durée : 15 minutes

NOM :

PRENOM :

NUMERO ETUDIANT :

Documents et téléphones interdits. Répondez dans l'emplacement prévu à cet effet.

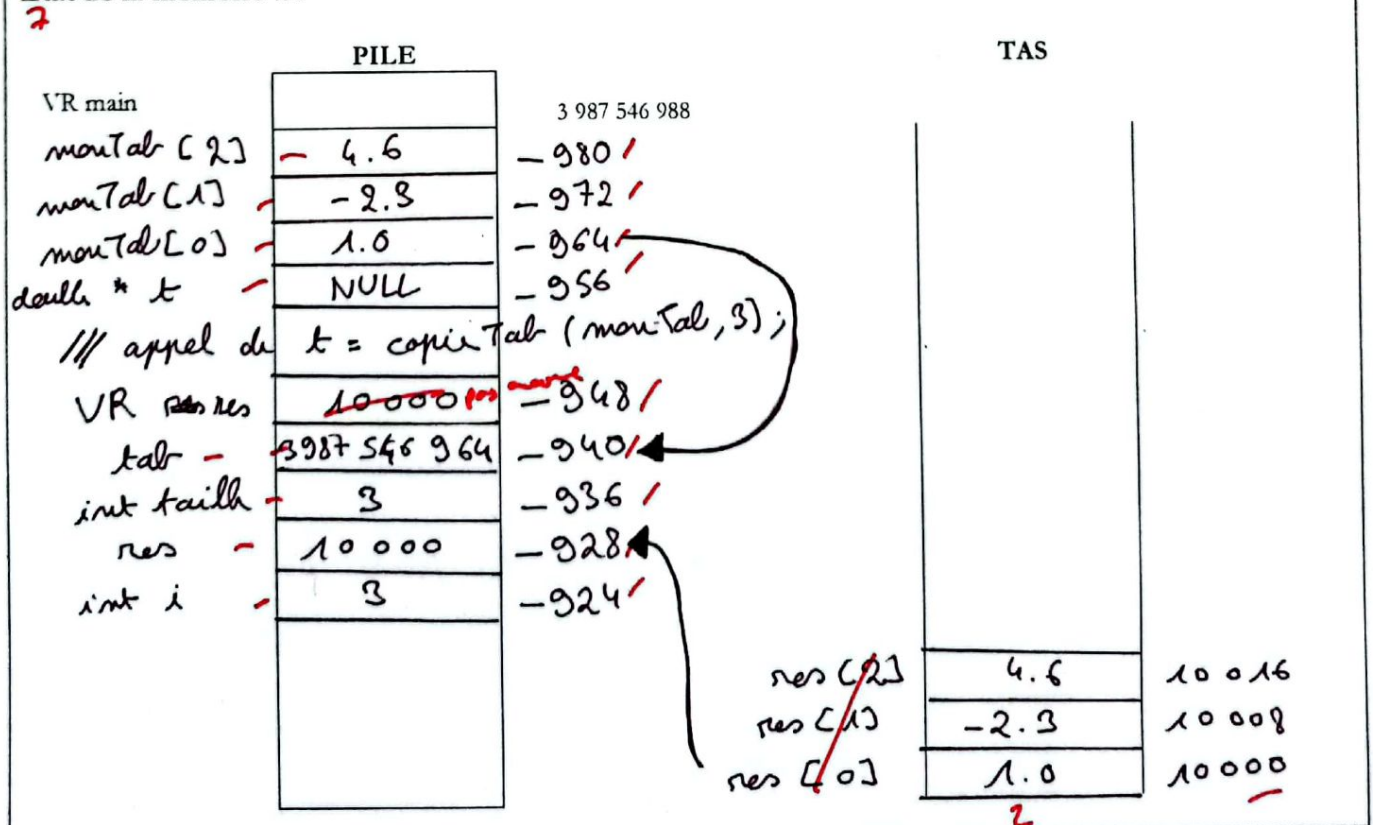
Soit le programme C++ suivant. Dessinez l'état de la mémoire aux endroits indiqués en commentaires. Vous utiliserez le modèle théorique de pile vu en cours et en TD. Vous supposerez que la valeur de retour du main est stockée à l'adresse 3 987 546 988 et qu'il n'y a pas de problème d'allocation mémoire.

```

1  double * copieTab (const double * tab, int taille) {
2      double * res = new double [taille];
3      int i;
4      for (i = 0; i < taille; i++) {
5          res[i] = tab[i];
6      }
7      /* Dessinez l'état de la mémoire #1 */
8      return res;
9  }
10
11 int main () {
12     double monTab [3];
13     monTab[0] = 1.0; monTab[1] = -2.3; monTab[2] = 4.6;
14     double * t = NULL;
15     t = copieTab(monTab,3);
16     /* Dessinez l'état de la mémoire #2 */
17     delete t;
18     return 0;
19 }

```

État de la mémoire #1



5

LIFAPSD - Evaluation de TD #2 - Groupes ABCD - sujet B

Date : lundi 7 novembre 2021 - Durée : 10 minutes

NOM :

PRENOM :

NUMERO ETUDIANT :

Documents et téléphones interdits. Répondez dans les emplacements prévus à cet effet.

Soit la fonction `contientQuestion` suivante qui teste et retourne si un texte de taille `n` passé en paramètre contient une question (en regardant s'il contient le caractère '?').

```
bool contientQuestion (const string & texte, int n) {
1  bool q = false;
2  int i = 0;
3  while (i < n) {
4      if (texte[i] == '?')
5          q = true;
6      i = i + 1;
7  }
8  return q;
}
```

1. Comptez le nombre d'opérations élémentaires effectuées dans le pire des cas en fonction de `n` pour chaque instruction de la fonction. **2**
- cas où le texte contient que des '?'*

Ligne	affectation entier	affectation booléen	comparaison entier	comparaison caractère	addition entier	indexation ([i])	return
1		1 ^{0.5}					
2	1 ^{0.5}						
3			$n+1$				
4				$n-1$		$n-1$	
5		$n-1$					
6	$n-1$				$n-1$		
7							
8							1 ^{0.5}
Total	n	n	n	$n-1$	$n-1$	$n-1$	1 ^{0.5}

2. En supposant que chaque opération élémentaire a un temps d'exécution d'une unité de temps, donnez la complexité en notation O de la fonction `contientQuestion`. **1**

$$O(\text{contientQuestion}) = 3n + 3(n-1) + 1 = 3n + 3n - 3 + 1 = 6n - 2$$

$$O(\text{contientQuestion}) = n$$

3. Compléter l'invariant ci-dessous de la boucle `while` des lignes 3 à 7 qui permet de prouver la correction de notre fonction (c'est-à-dire qu'elle retourne si le texte contient une question). **2**

« Au début de toute itération i , la variable ^{0.5} q indique s'il existe ^{0.5} le caractère $'?'$ dans le texte entre les indices 0 et $i-1$. »

Handwritten notes: $i < n$ and $i-1$ are circled and annotated with 0.5.