

### Exercice I

Donner l'état de la mémoire ainsi que le résultat de l'exécution du programme suivant :

```
#define N 5
typedef struct Nœud{
    int d;
    char *c;
} nœud;

void f(nœud *a, int i, char *pc){
    (a+i)->d=i;
    (a+i)->c=pc;
}

void main(){
    char *cours="INFO216";
    int t[N]={1,2,3,4,5}, i;
    nœud tn[N];
    for(i=0;i<N;i++){
        f(tn,i,cours);}
}
```

### Exercice II

La fonction *strpbrk()* de la bibliothèque C recherche dans *str1* la première occurrence de n'importe quel caractère de *str2* et retourne la sous chaîne de *str1* débutant par cette occurrence si elle est trouvée, et NULL sinon.

Voici la déclaration de la fonction *strpbrk()*: `char *strpbrk(char *str1, char *str2)`

L'exemple suivant illustre l'utilisation de la fonction *strpbrk ()* :

```
#include <stdio.h>
#include <string.h>

int main () {
    char str1[] = "abcde2fghi3jk4l";
    char str2[] = " xhyf0b";
    char *ret = strpbrk(str1, str2);
    if(ret) {
        printf("La sous chaine de correspondance: %s\n", ret);}
    else {
        printf("Aucun caractere de %s n'est trouvable dans %s", str2, str1);}
    return(0);
}
```

Ce qui produira le résultat suivant: `sous chaine de correspondance: bcde2fghi3jk4l`

Il vous est demandé d'écrire votre propre fonction *strpbrk ()*, sans utiliser les fonctions de *string.h* (même pas la fonction *strlen*).

### Exercice III

Une entreprise qui produit des articles, gère les commandes de ses clients par une liste simplement chaînée. Une commande est définie par le nom de client faisant la commande, le code d'article commandé (chaîne de caractères) et la quantité commandée

1. Définir en C la structure récursive représentant une commande.
2. Écrire la fonction récursive *qtItem* qui, ayant en entrée la liste de commandes et un code d'un article, calcule et retourne la quantité totale commandée de cet article.