

TD 5 - Algorithmique et programmation Arbres

Exercice 1

```
1  #include <stdio.h>
2  struct integerChain_ {int value ; struct integerChain_* next ; } ;
3  typedef struct integerChain_* integerChain ;
4
5  integerChain createNew (int value ) {
6      integerChain new ;
7      new = ( integerChain ) malloc (sizeof(struct integerChain_ ) ) ;
8      new->value = value ;
9      new->next = NULL ;
10     return new ;
11 }
12
13 int main (int argc, char** argv ) {
14     int * a = (int *)calloc(16, sizeof(int));
15     //A
16     for (int i = 1; i<16; i++){a[i-1]>5? a[i]=i : a[i]-=i;}
17     //B
18     int temp=0;
19     int positifs=0;
20     for (int i = 1; i<16; i++){a[i]>0? positifs++;}
21     int * b = (int *)calloc(positifs, sizeof(int));
22     //C
23     for (int i = 0; i<16; i++)
24     {
25         if (a[i]>0){
26             b[temp]=a[i];
27             temp++;
28         }
29     }
30     free(a); free(temp);
31     //D
32     integerChain start = createNew(0);
33     integerChain temp = start;
34     for (int i = 1; i<positifs; i++)
35     {
36         temp->value = b[i];
37         temp->next = createNew(0);
38         temp=temp->next;
39     }
40     //E
41     return 0;
42 }
```

On suppose que l'adressage commence à 0x1000 et se poursuit sans interruption. On se place dans un milieu où $\text{sizeof}(\text{int}) = \text{sizeof}(\text{int}^*) = 4$. Pour chaque marqueur A, B, C, D et E, indiquer l'état de la mémoire attribué aux différentes variables du programme.

Exercice 2

Developper deux fonctions de prototype "int nom(int a)" qui respectivement multiplie ou divise l'argument d'entrée par 10 puis le renvoie (et l'affiche à l'écran).

Ecrire une fonction qui, au travers de l'utilisation de pointeurs, appelle la fonction de division si un nombre est un multiple de 10, celle de multiplication sinon.

Exercice 3

On souhaite écrire une batterie de tests à effectuer sur une chaîne de caractères. Ces tests renvoient tous 0 (test échoué) ou 1 (test réussi).

1. Presence de majuscules
2. Nombre de 'e' supérieur à 100
3. Existence d'un a suivi d'un n
4. Aucun chiffre

Pour chaque test, écrire la fonction correspondante. A l'aide d'une syntaxe telle que : int (*p[4]) (char* chaine), créer un tableau de pointeurs vers ces fonctions, puis les appliquer à un texte de votre choix dans une boucle (un article wikipédia ou le lorem ipsum par exemple).