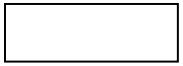
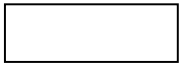
I2010 : langage C (TP5)

Pointeurs et gestion dynamique de la mémoire

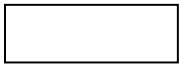
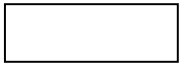
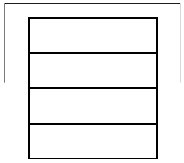
1. Exercice de compréhension

Ci-dessous quelques déclarations et instructions, après chaque instruction, complétez le dessin pour montrer le contenu des variables.

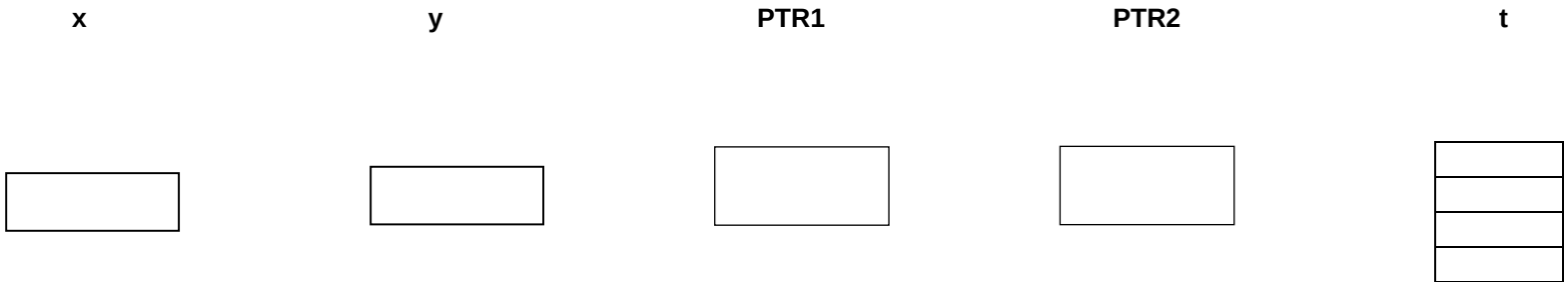
```
int main()
{
	int x = 1;
	int y = 20;
	int t[4] = {3, 4};
	int *ptr1, *ptr2;

	ptr1=&x;
	ptr2=t;
```

x	y	PTR1	PTR2	t
				

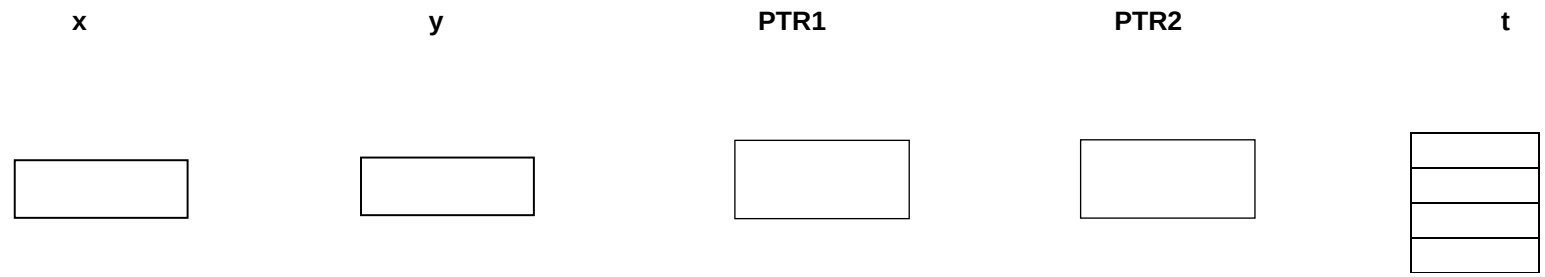
```
	y=(*ptr1)++;
```

x	y	PTR1	PTR2	t
				

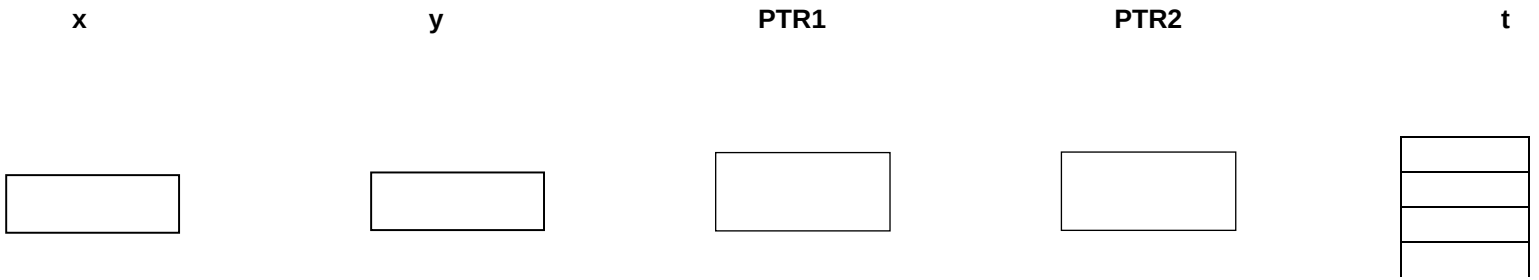
```
ptr2++;
```



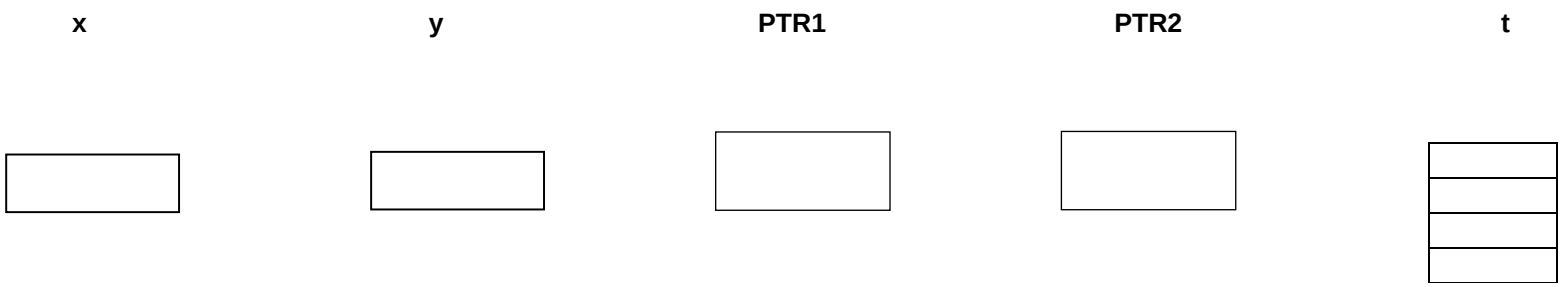

```
*ptr2 = *ptr1;
```



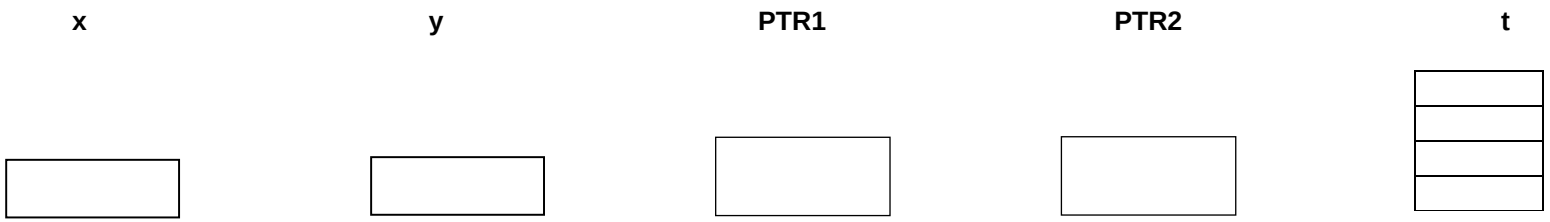

```
ptr1 = ptr2 + *ptr2;
```



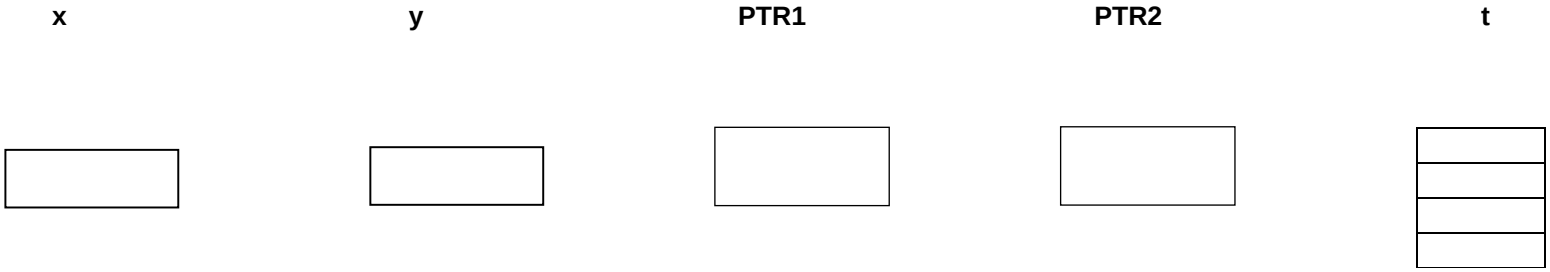
```
*ptr1 = *(ptr2-y);
```




```
ptr1 = &(t[x]);
```

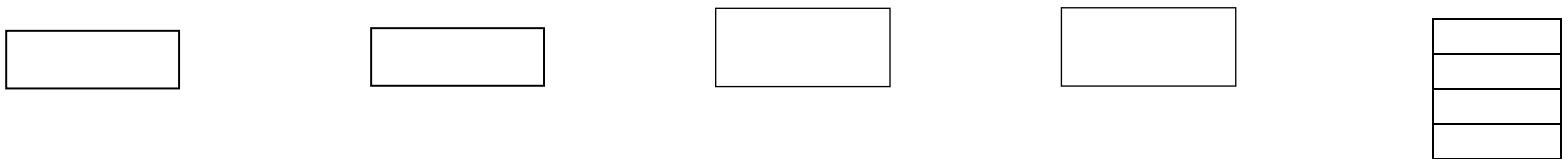



```
*ptr1 = y;
```



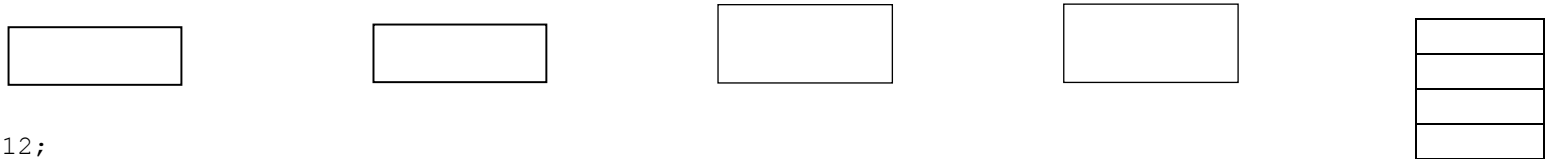
```
*ptr2 = *ptr1--;
```

x **y** **PTR1** **PTR2** **t**



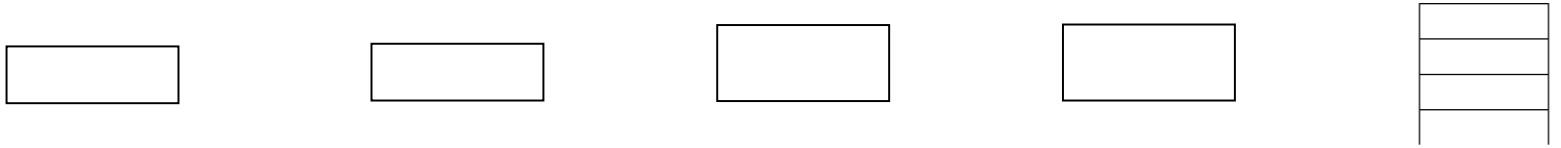
```
ptr2 = t + 3;
```

x **y** **PTR1** **PTR2** **t**



```
*ptr2 = 12;
```

x **y** **PTR1** **PTR2** **t**



x **y** **PTR1** **PTR2** **t**

```
y = ptr2-ptr1;
```

```
return 0;
}
```

2. Fuite mémoire

Compiler et exécutez le programme *mem_leak.c*
Lancez le process en tâche de fond avec la commande *./a.out &*
Ensuite observez l'évolution de l'occupation mémoire grâce à la commande *top*
Quel est le problème ?
Quelle divergence voyez-vous par rapport au *man* de la fonction *malloc* ?
Corrigez ce code.

Exercices de programmation des pointeurs

3. Allocation dynamique de tableaux à une dimension

Ecrivez un programme qui lit sur *stdin* :

- un entier `n` qui représente le nombre de données ;
- `n` entiers qui peuvent être soit positifs, nuls ou négatifs.

Après avoir lu les données, le programme créera et affichera deux tableaux :

- l'un contiendra la liste des entiers >=0 ;
- l'autre la liste des entiers <0

Exemple :

Input
```
5 -2 56 12 -3
```

Output
```
5 56 12
-2 -3
```

Après avoir traité les entrées et affiché le résultat, le programme redemandera un nouveau jeu de données à traiter tant que `n>0`.

Utilisez la fonction *scanf*.

Vous devez allouer dynamiquement les tableaux de sorte que tous les éléments soient assignés (i.e. taille logique = taille physique) et libérer la place qu'ils occupent après leur utilisation.

4. Arithmétique des pointeurs

Pour rappel, lorsque vous avez un pointeur dans un tableau, vous pouvez passer à l'élément suivant en incrémentant ce pointeur (via l'opérateur ++).

Pour passer d'une version indicée à une version pointeurs de votre programme, modifiez-le afin de ne plus utiliser l'opérateur **[ ]**.