

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 = 6;
    int y = 20;
    int z = '5';
    int *ptr1, *ptr2;

    ptr1=&x;
    ptr2=&z;
```

PTR1

PTR2

X

Y

Z

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

PTR1

PTR2

X

Y

Z

```
*ptr2=='0';
```

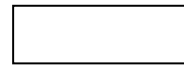
PTR1

PTR2

X

Y

Z



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

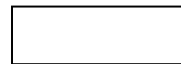
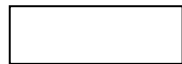
PTR1

PTR2

X

Y

Z



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

PTR1

PTR2

X

Y

Z



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

PTR1

PTR2

X

Y

Z

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

PTR1

PTR2

X

Y

Z

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

PTR1

PTR2

X

Y

Z

```
ptr1=&y;
```

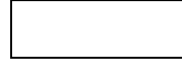
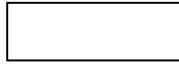
PTR1

PTR2

X

Y

Z



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

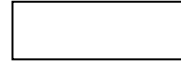
PTR1

PTR2

X

Y

Z



```
ptr2 = &x ;  
*ptr1=*ptr2+1;
```

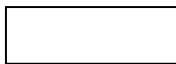
PTR1

PTR2

X

Y

Z



```
ptr2 = ptr1 ;
```

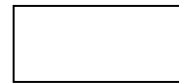
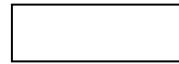
PTR1

PTR2

X

Y

Z



```
return 0;
```

```
}
```

2. Exercices de programmation des pointeurs

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

Vous devez allouer dynamiquement les tableaux et libérer la place qu'ils occupent après leur utilisation.

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 [].