

TD 2

LES POINTEURS

Exercice 1

On considère les cinq variables suivantes : `int A = 1, B = 2, C = 3, *P1, *P2`. Compléter le tableau en indiquant les valeurs des différentes variables au terme de chaque instruction du programme suivant :

Instruction	A	B	C	P1	*P1	P2	*P2
<code>P1 = &A</code>							
<code>P2 = &C</code>							
<code>*P1 = (*P2) ++</code>							
<code>P1 = P2</code>							
<code>P2 = &B</code>							
<code>*P1-- = *P2</code>							
<code>++*P2</code>							
<code>*P1* = *P2</code>							
<code>A = ++*P2**P1</code>							
<code>P1 = &A</code>							
<code>*P2 = *P1 / *P2</code>							

Exercice 2

Soit un pointeur `p` qui pointe vers un tableau `A` comme suit :

```
int A[] = {12,23,34,45,56,67,78,89,90};
```

```
int *p;
```

```
p = A;
```

Quelles valeurs ou adresses fournissent ces expressions sachant que `&A[0]=bfb8ffc0` :

Q1. `*P+2`

Q2. `*(P+2)`

Q3. `&P`

Q4. `&P+1`

Q5. `&P+2`

Q6. `&A[2]`

Q7. `P+3`

Q8. `&A[7]-P`

Exercice 3

Quels résultats fournira ces deux programmes :

programme 1 :

```
main()
{
  int t[4]={10,20,30,40};
  int * ad[4];
  int i;
  for(i=0;i<4;i++) ad[i]=t+i; /*1*/
  for(i=0;i<4;i++) printf("%d" , *ad[i]); /*2*/
  printf("\n");
  printf("%d %d \n",*(ad[1]+1),*ad[1]+1); /*3*/
}
```

Programme 2 :

```
main()
{
  char *ad;
  ad="bonjour"
  printf("%s\n",ad);
  ad="monsieur"
  printf("%s",ad);
}
```

Exercice 4

Ecrire une fonction `void affiche_entiers(int* tab,int nb)` qui prend en argument un tableau d'entier et le nombre d'éléments qu'il contient et qui affiche ces éléments. N'utilisez pas la notation `tab[i]` pour parcourir le tableau, mais plutôt l'opérateur `++`.

Exercice 5

L'objectif de cet exercice est d'implémenter la fonction `strcmp` qui permet de comparer deux chaînes de caractères `s1` et `s2`. Compléter le code suivant afin de répondre à cet objectif

```
int strcmp(char *s1,char *s2)
{
  while (*s1 != '\0' && *s1 == *s2)
  {
    .....;
    .....;
  }
  if (*s1<*s2)
  return (...);
else
  return (.....);
}
```