

# Evaluation n°1

Nom :

Prénom :

Scalaires, pointeurs, fonctions  
chaînes, bitwises, tableaux 1D

Quelles sont les valeurs affichées lors de l'exécution de la fonction suivante ?



```
void scalaires()
{
    printf("%c %d\n", 'a'+5, sizeof('8'));

    int a,b=0;
    for(a=3; a>=0; a--)
    {
        printf("(%d %d %d)\n", a!=b, a&&b, !a);
        b++;
    }
}
```

A1 R P A

A2 R P A

A5 R P

Quelles sont les valeurs affichées lors de l'exécution de la fonction suivante ?



```
void pointeurs()
{
    int x=10,y=20;
    int* p=&x;
    int* q=&y;
    (*p)++; p=q; (*p)++; (*q)++;
    printf("%d %d %d %d\n", x,y,*p,*q);
}
```

A3 R P A

Expliquez de façon simple et concise ce que réalise cette fonction.

```
int* f1(int* p, int* q)
{
    if(*p<*q) return p;
    else return q;
}
```

A4 R P A



Quelles est la valeur affichée lors de l'exécution de la fonction suivante ?



```
void fpointeurs()
{
    int a=9,b=7;
    int* p=f1(&a,&b);
    printf("%d\n", *p);
}
```

A4 R P

Quelles sont les valeurs affichées lors de l'exécution de la fonction `bitwises` ?

```
void f3(unsigned char* x)
{
    unsigned char m = 1 << 7;
    *x = *x >> 1;
    *x = *x | m;
}
```

```
void bitwises()
{
    unsigned char w=0x0F;
    int i;
    for(i=0; i<4; i++)
    {
        printf("%x ",w);
        f3(&w);
    }
}
```

A4	R	P	A
----	---	---	---

B1	R	P	A
----	---	---	---



Quelles sont les valeurs affichées lors de l'exécution de la fonction `tableaux` ?

```
void tableaux()
{
    double t[6] = {1.0,2.0,3.0,4.0,5.0,6.0};
    double* p = t+2;
    f4(p,3);
    f5(t,6);
}
```

```
void f4(double t[], int n)
{
    int i;
    for(i=0; i<n; i++)
        t[i]=0.0;
}
```

```
void f5(double t[], int n)
{
    int i;
    for(i=0; i<n; i++)
        printf("%lf ",t[i]);
}
```



B3	R	P	A
----	---	---	---

On souhaite représenter un ensemble `E` d'entiers pouvant contenir des valeurs comprise entre 0 et 31 par un entier. Le principe de cette représentation très concise est que si `E` contient la valeur `i`, alors le bit de rang `i` de l'entier qui représente `E` vaut 1, sinon il vaut 0. Donnez le code de la fonction `add` qui ajoute un élément à un ensemble et de la fonction `printSet` qui affiche les valeurs contenues dans un ensemble.

```
void add(int* set, int x)
{

```



```
void printSet(int* set)
{

```



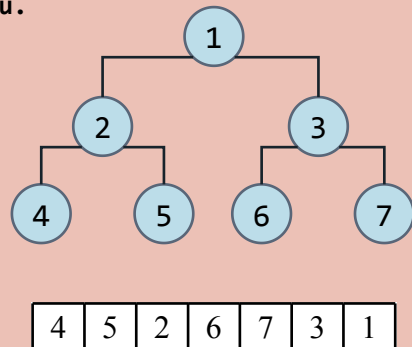
B1	P	A	C
----	---	---	---

On souhaite représenter sous la forme d'un tableau d'entiers un arbre binaire complet dont les nœuds sont étiquetés par des entiers.

Un arbre de profondeur 1 possède un seul nœud et est représenté par un tableau de une cellule contenant la valeur associée à ce nœud.

Un arbre de profondeur  $n > 1$  est représenté par un tableau contenant la représentation de son fils gauche, puis celle de son fils droit, puis la valeur associée à sa racine.

Un exemple d'arbre binaire complet de profondeur 3 et sa représentation par tableau.



La fonction `filsgauche`, qui retourne un pointeur sur la représentation du fils gauche d'un arbre, vous est donnée gratuitement.

Complétez les définitions des fonctions `filsdroit` et `racine` qui retournent respectivement un pointeur sur le fils droit et la valeur de la racine d'un arbre de profondeur donnée.

```
int* filsgauche(int* arbre, int prof)
{
    return arbre;
}
```

Gratuit

```
int racine(int* arbre, int prof)
{
}

```



```
int* filsDroit(int* arbre, int prof)
{
}

```



Complétez la définition de la fonction `printInfixe` qui affiche *en ordre infixe* les valeurs associées aux nœuds d'un arbre binaire complet.

Pour mémoire, l'affichage en ordre infixe consiste à afficher d'abord le fils gauche en ordre infixe, puis la valeur de la racine, puis le fils droit en ordre infixe.

Pour l'arbre d'exemple, cela donne 4 2 5 1 6 3 7.

Utilisez les fonctions `racine`, `filsdroit` et `filsgauche`.

```
void printInfixe(int* arbre, int prof)
{
}

```



Réalisez une fonction `conv` qui accepte en paramètre une chaîne `s` supposée contenir uniquement des caractères '0' et '1' et qui retourne l'entier dont `s` est la représentation en base 2.

Par exemple, `conv('11010')` doit retourner 26.

```
int conv(char* s)
{

}
}
```



B4	P	A	C
----	---	---	---