

## TD 4

### Pointeurs, pile d'appel et traces de programmes

Réaliser les traces des programmes suivants

#### EXERCICE 1 - Appel de fonction

```
#include<stdio.h>
#include<stdlib.h>

int max_valeur_absolue(int x, int y){
    if( x < 0 )
        x = -x;
    if( y < 0 )
        y = -y;
    if( x < y )
        return y;
    return x;
}

int main(){
    int x = -4;
    int y = 2;
    int z = max_valeur_absolue(x, y);
    return EXIT_SUCCESS;
}
```

#### EXERCICE 2 - Swap

```
#include <stdio.h>
#include <stdlib.h>

void swap(int * x, int * y){
    int tmp = *x;
    *x = *y;
    *y = tmp;
}

int main(){
    int x = 3;
    int y = 4;
    swap(&x, &y);
    return EXIT_SUCCESS;
}
```

#### EXERCICE 3 - Puissance, version itérative

---

```
#include<stdio.h>
#include<stdlib.h>
```

```
int puissance_iter(int x, int n){
    int resultat = 1;
    int i;
    for( i = 0; i < n; i++)
        resultat *= x;
    return resultat;
}
```

```
int main(){
    int res = puissance_iter(2, 4);
    printf("2^4 = %d\n", res);
    return EXIT_SUCCESS;
}
```

EXERCICE 4 - Puissance, version récursive

```
#include<stdio.h>
#include<stdlib.h>
```

```
int puissance_rec(int x, int n){
    if( n == 0 )
        return 1;
    return x * puissance_rec(x, n - 1);
}
```

```
int main(){
    int res = puissance_rec(2, 4);
    printf("2^4 = %d\n", res);
    return EXIT_SUCCESS;
}
```

EXERCICE 5 - Tableaux

```
#include<stdio.h>
#include<stdlib.h>
```

```
int * init_tableau(int n, int val){
    int * resultat = malloc(sizeof(int) * n);
    int i;
    for(i = 0; i < n; i++)
        resultat[i] = val;
    return resultat;
}
```

```
int main(){
    int * tab = init_tableau(4, -1);
    free(tab);
    return EXIT_SUCCESS;
}
```

---

```
}
```

## EXERCICE 6 - Pointeur de fonctions

```
#include<stdio.h>
#include<stdlib.h>
```

```
int addition(int x, int y){
    return x+y;
}
```

```
int soustraction(int x, int y){
    return x-y;
}
```

```
int main(){
    int (*operation)(int, int) = addition;
    int x = operation(3,4);
    operation = soustraction;
    int y = operation(x, 2);
    printf(" Resultat : %d\n",y);
    return EXIT_SUCCESS;
}
```