Tutorial 7 : Pointers

Exercise 1:

Complete the following table:

C code	a	b	c	p1	*p1	p2	*p2
int a, b, c, *p1 , *p2 ;							
a = 1, b = 2, c = 3;							
p1 = &a, p2 = &c;							
*p1 = (*p2)++;							
p1 = p2 ;							
p2 = &b;							
*p1 −= *p2;							
++*p2;							
*p1 *= *p2;							
a = ++*p2 * *p1;							
p1 = &a;							
*p2 = *p1 / = *p2 ;							

Exercise 2:

Let iptr be a pointer that points to an array tabA:

```
int tabA[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int *iptr=NULL;
iptr = tabA;
```

What values or addresses do these expressions provide:

1. *iptr+2 ,
2. *(iptr+2),
3. &iptr+1 ,
4. &tabA[4]-3 ,
5. tabA+3 ,
6. &tabA[7]-iptr
7. iptr+(*iptr-10) ,
8. *(iptr+*(iptr+8)-tabA[7])

Exercise 3:

1. Write a C function that takes two real-type values x and y and returns the maximum and the difference between the two numbers.

2. Use your function in a main program with two numbers entered by the user.

Exercise 4:

1) for: int x[5] = {0, 1, 2, 3, 4} ;

what will be the values of x after the call modify1(x, 1, 4) ?

```
void modify1(int a[], int i, int j) {
   int t;
   t = a[i]; a[i] = a[j]; a[j] = t;
}
```

2) for: int x[5] = {0, 1, 2, 3, 4} ;

What will be the values of x after the call modify2(x+1, x+4) ?

```
void modify2(int *pa, int *pb) {
   int t;
   t=*pa;  *pa=*pb;  *pb=t;
}
```

3) for: int x[5] = {0, 1, 2, 3, 4} ;

What is the result of display(&x[0]) ? display(&x[2]) ? display(&x[4]) ?

```
void display(int x[]) {
   int i;
   for (i = 0; i < 3; i++)
         printf("%d ", x[i]);
}
```

Exercise 5:

Write a C program to reverse a vector of N elements (N given by the user). Use pointers to iterate through the elements of the vector.