

## Practical Work Number 3

### C Programming - Pointers

#### Part 1

1- Consider the following program. Complete this program by displaying the memory addresses of i, j, p1, and p2. Then, print the values of i, j, p1, and p2.

```
#include<stdio.h>
int i,j;
int *P1,*P2;
int main(){
i=2; j =5;
P1=&i; P2=&j;
return 0;}
```

2- By adding the following instruction, what is the result of the execution?

```
(*P1)++;
*P2=*P1;
i=18;
*P1/=*P2;
j=++(*P1);
j=(*P1)++;
printf("i=%d\n",i);
printf("j=%d\n",j);
printf("*P1=%d\n",*P1);
printf("*P2=%d\n",*P2);
```

#### Part 2 (exercise)

Let P be a pointer pointing to the array t:

```
int t[]={1,3,14,16 ,20 ,90,10,3,60};
int *p;
p=t;
```

What values or addresses do these expressions provide?

- 1) p
- 2) \*p
- 3) \*(p+2)
- 4) p+3
- 5) &t[5]-3
- 6) &t[7]- &t[5]
- 7) \*(p+\*(p+8)-t[7])

### **Part 3**

Write a program that allows reading an array of integers of size N entered from the keyboard:

- 1- Display the memory address and the content of each element of the array.
- 2- Display the even numbers from the array.

Write a program that allows reading a string entered from the keyboard:

- 1- Display the memory address and the content of each character of the string.
- 2- Display the number of occurrences of the character 'C'.

**Note:** Traversing the array and the string should be done using pointers.