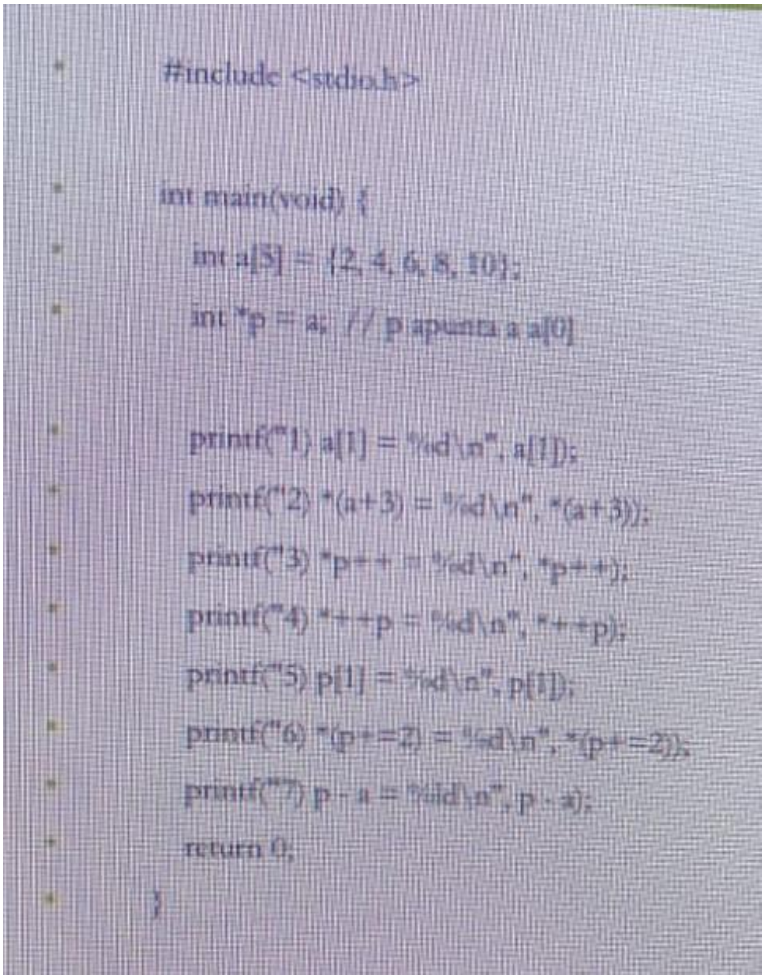


```
#include <stdio.h>

int main(void) {
    int a[5] = {2, 4, 6, 8, 10};
    int *p = a; // p apunta a a[0]

    printf("1) a[1] = %d\n", a[1]);
    printf("2) *(a+3) = %d\n", *(a+3));
    printf("3) *p++ = %d\n", *p++);
    printf("4) *++p = %d\n", *++p);
    printf("5) p[1] = %d\n", p[1]);
    printf("6) *(p+=2) = %d\n", *(p+=2));
    printf("7) p - a = %d\n", p - a);
    return 0;
}
```

La línea 3 donde empieza lo bueno **declara un arreglo de 5 enteros con los valores 2, 4, 6, 8 y 10**

**Línea 4 ya nos dice que hace xd**

Línea 5 (de código no la vacía)Muestra el valor del segundo elemento del arreglo a[1] que es **el valor 4**

Linea 6 Usa al puntero a+3  nuestro primer puesto más 3 dándonos el valor de nuestra cuarta **posición que es  8.**

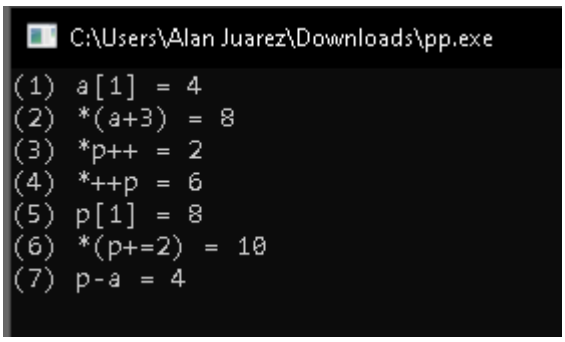
Linea 7 *p++ obtiene el valor apuntado por p (“2), luego incrementa p para que apunte a nuestro **valor que seria 2**

Linea 8 *++p primero incrementa p (ahora apunta a a[2]), luego obtiene el **valor de 6**

Linea 9 p actualmente apunta a a[2] que es 6, por lo que p[1] es como sumarle 1 más al arreglo volviéndolo a[3] lo que nos va a arrojar el **valor 8**

Linea 10 p+=2 apunta al arreglo a[4] (desde a[2] + 2 = a[4]), que tiene el **valor de 10**

Linea 11 aquí todavía nuestro apuntador marca que esta en a[2]=6 pero ahora se le está quitando una posición por el p-a por lo que nos queda **a[1]=4**



```
C:\Users\Alan Juarez\Downloads\pp.exe
(1) a[1] = 4
(2) *(a+3) = 8
(3) *p++ = 2
(4) *++p = 6
(5) p[1] = 8
(6) *(p+=2) = 10
(7) p-a = 4
```