

TRABAJO PRÁCTICO N° 5

Deben entregarse por grupos en la entrega correspondiente vía Campus los archivos .c correspondientes a los ejercicios indicados.

1. Asumiendo que $x = 5$, $y = 10$, ¿Cuál es el valor de x e y después de llamar a `swap1(x, y)`?

```
void swap1(int a, int b)
{
    int t;

    t = a;
    a = b;
    b = t;
}
```

2. Asumiendo que $x = 5$, $y = 10$, ¿Cuál es el valor de x e y después de llamar a `swap2(&x, &y)`?

```
void swap2(int *pa, int *pb)
{
    int t;
    t = *pa;
    *pa = *pb;
    *pb = t;
}
```

3. Asumiendo que $x[5] = \{0, 1, 2, 3, 4\}$. ¿Cuáles son los valores de x después de llamar `swap3(x, 1, 4)`?

```
void swap3(int a[], int i, int j)
{
    int t;
    t = a[i];
    a[i] = a[j];
    a[j] = t;
}
```

4. Asumiendo que $x[5] = \{0, 1, 2, 3, 4\}$. ¿Cuáles son los valores de x después de llamar `swap2(x + 1, x + 4)`?

5. Asumiendo que `x[5] = {0, 1, 2, 3, 4}`. ¿Qué imprime `print3(&x[0])`? ¿Y `print3(&x[2])`? ¿Y `print3(&x[4])`?

```
void print3(int *a)
{
    printf("%d\n", *a);
}
```

6. Cuando se pasa un *array* a una función de C, en realidad se pasa un puntero al primer elemento. ¿Por qué C no crea una copia local del *array*, tal como hace con números enteros?

7. ¿Cuál es la diferencia entre éstos dos programas?

```
int middle1(int a[], int n)
{
    return a[n/2];
}
```

```
int middle2(int *a, int n)
{
    return a[n/2];
}
```

8. ¿Qué tiene de malo el siguiente programa?

```
#include <stdio.h>

void set(int *x)
{
    *x = 10;
}

int main(void)
{
    int *x;
    set(x);
    printf("%d\n", *x);
    return 0;
}
```

9. ¿Qué ocurre si declaramos un array de la siguiente manera?

```
#define      N      100

int a[N * N];
```

10. Escribir una función que intercambie dos números usando punteros, que tenga el siguiente prototipo:

```
void swap(int * a, int * b)
```

11. **[ENTREGAR]** Escribir un programa que implemente el algoritmo de ordenamiento *bubblesort*¹ a partir de la función `swap()` escrita en el ejercicio anterior. El programa deberá ordenar una lista de números enteros separados por comas ingresada por el usuario. El programa debe imprimir la lista ordenada, y reportar al usuario cuantos intercambios se realizaron en total para ordenar la lista.

Un ejemplo de como podría ser la salida del programa:

```
Ingrese la lista de elementos a ordenar: 10, -3, 4

Lista ordenada: -3, 4, 10

Se debieron realizar 2 intercambios para ordenar la lista.
```

Recuerden validar los datos que ingresa el usuario de manera de que el programa se comporte adecuadamente ante entradas inválidas.

¹ https://es.wikipedia.org/wiki/Ordenamiento_de_burbuja
Trabajo Práctico N° 5