



Práctica 2 – Punteros y Memoria Dinámica

1. Corregir el siguiente programa para que los valores de las variables *a* y *b* resulten ordenados de manera ascendente:

```
#include <stdio.h>

int main(){
    int a = 30, b = 20;
    ordenadas(a, b);
    printf(" valor de a %d\tvalor de b %d\n", &a, &b);
    return 0;
}

void ordenadas(int x, int y){
    int* aux;
    if(x > y) {
        aux = x;
        x = y;
        y = aux;
    }
}
```

2. Describir lo que imprime el siguiente fragmento de código:

```
int *p, a = 4; b = 5;
p = &b;
*p *= 2;
printf("b=%d *p=%d\n", b, *p); _____
printf("&b=%p p=%p &p=%p\n", &b, p, &p); _____
b = *p *3;
printf("b=%d *p=%d\n", b, *p); _____
printf("&b=%p p=%p\n", &b, p); _____
a = b;
p = &a;
(*p)++;
printf("b=%d a=%d *p=%d\n", b, a, *p); _____
printf("&b=%p &a=%p p=%p &p=%p\n", &b, &a, p, &p); _____
```

3. Implementar un programa que cree dinámicamente 3 variables enteras, muestre su suma y su producto. Asegurarse de administrar correctamente la memoria e implementar una función para evitar duplicaciones de código en la creación y lectura de cada variable.
4. Desarrollar un programa que cree dinámicamente un arreglo de números reales que contenga *N* elementos (*N* es ingresado por teclado). Ingresar sus elementos y mostrar aquellos que sean positivos utilizando aritmética de punteros. Al finalizar, liberar la memoria solicitada en tiempo de ejecución.
5. Desarrollar un programa que cree un arreglo estático de punteros a enteros, y luego cargue en él una cantidad desconocida de enteros (se encuentran en un archivo de texto). Mostrar aquellos que sean positivos. Al finalizar, liberar la memoria solicitada en tiempo de ejecución.