Práctica de punteros (habilidad a,b)

1) Qué se imprime?

```
#define SIZE 10
  ……
int array[SIZE];
int *ptr = array; // This is the same as int *ptr = &array[0];
for (int i = 0; i < SIZE; i++) {
     *(ptr + i) = i + 1;
}
 for (int i = 0; i < SIZE; i++) {
    cout << array[i] << " ";
}
```

Nota: Los arreglos no son enviados a las funciones como parámetros, sino un puntero al primer elemento.

2) Dada la función swap

```
void swap (int& x, int& y) {
      int temp = x;
      x = y;
      y = temp;
 }
```

Qué se imprime?

```
int a = 5, b = 10, c = 7;
swap(a, b);
swap(a, c);
cout << a << " " << b << " " << c << endl;
```

Práctica de punteros (individual)

3) Qué se imprime?

```
char a = 'L', b = 'T';
char *p1 = &a, *p2 = &b, *p3;

p3 = &b;
cout << *p3 << endl;
p3 = p1;
cout << *p3 << endl;
*p1 = *p2;
cout << *p1 << endl;
```

4) Dadas las siguientes variables

```
int *p, i = 32, k = i;
p = &i;
```

Qué opción actualiza el valor de i a 45

1. k = 45
2. *k = 45
3. p = 45
4. *p = 45
5. Dos o más respuestas

Práctica de punteros (individual)

5) Da error? Si da error explica.

```
char c = 'A';
double *p = &c;
```

6) Implementar una función swap similar a la del ejercicio 2, pero con la siguiente firma:

```
void swap (int* x, int* y);
```

Da un ejemplo de como usuarias la función en el main:

7) Qué imprime este código?

```
int a;
void funct(int x, int **y) {
        a = 10;
        **y = x - a;
        *y = &a;
}
int main(int argc, char const *argv[]) {
      int a, b = 8, *p = &b;
        a = 30 + b;
        p = &a;
        *p = 20 - b;
        funct (a, &p);
        *p = 3;
        cout << a << " " << b << endl;
}
```

Práctica de punteros (individual)

8) Qué imprime este código?

```
int ia[] = {0, 2, 4, 6, 8};
int *ptr = ia;
cout << *ptr << endl;

ptr = &ia[4];
cout << *ptr << endl;
```

9) Qué imprime este código?

```
int ia[] = {0, 2, 4, 6, 8};
int *ptr = ia;
int *ptr2 = ptr + 4;

cout << *ptr2 << endl;

int aux = *ptr2 + 7;
cout << aux << endl;
```

Práctica de punteros (individual)

10) Qué imprime este código?

```
int ia[] = {0, 2, 4, 6, 8};
int *ptr = &ia[3];
ptr = &ia[4];
cout << *ptr << endl;
```

11) Desarrolla un código que defina un entero “a” y un puntero a un entero “ptr” que guarde la dirección del entero “a”. Finalmente imprime el valor del entero “a”, su dirección de memoria y la dirección del puntero “ptr”.

12) Qué imprime este código?

```
int index, *pt1, *pt2;
index = 39;
pt1 = &index;
pt2 = pt1;
*pt1 = 13;

cout << index << “ “ << *pt1 << “ “ << *pt2 << endl;
```

Práctica de punteros (individual)

13) Dada la función swap

```
void swap (int x, int y) {
    int temp = x;
    x = y;
    y = temp;
}
```

Qué se imprime?

```
int a = 5, b = 10, c = 7;
swap(a, b);
swap(a, c);
cout << a << " " << b << " " << c << endl;
```

14) Qué imprime este código?

```
int x = 1, y = 2;
int *ip;
ip = &x;
y = *ip;
*ip = *ip + 3;

cout << x << " " << y << endl;
```

Práctica de punteros (individual)

15) Diseñe un programa, que sume dos variables de tipo entero, por medio de punteros.

16) Dada la función?

```
void increment(int *number) {
    *number = *number + 1;
}
```

Qué imprime este código?

```
int x = 18;
increment(&x);
cout << x << endl;
```

17) Qué imprime este código?

```
int* ptr;
*ptr = 5;

cout << *ptr << endl;
```

18) Qué imprime este código?

```
int x = 15;
int* ptr = &*&x;
x -= 7;
*ptr = x + 10;

cout << *&*ptr << endl;
```

Práctica de punteros (individual)

19) Qué hace la siguiente función?

```
char* func(char *str) {
        char *p;
        int len = strlen(str);
        p = &str[len-1];

        while (*p == ' ') {
              --p;
        }
        *(p + 1) = '\0';
        return str;
}
```

20) Qué hace la siguiente función?

```
char* func(char *str) {
        char *p;
        int len = strlen(str);
        p = str;

        for (int i=0; i < len; ++i) {
              if (*p == ',') {
                    *p = ' ';
              }
              ++p;
         }
         return str;
}
```