

Estructura de Datos y Algoritmos

Unidad 01: Crecimiento de funciones

Todos los profesores

1 Ejercicio 01

Suma y Promedio de números.

```
int s = 0;
double prom = 0;
for (int i=0; i<n; i++) {
    s = s + A[i];
    prom = s / (double)n;
}
printf("Suma:%d\nProm:%d", s, prom);
```

2 Ejercicio 02

Suma y Promedio de números (2.0).

```
int s = 0;
double prom = 0;
for (int i=0; i<n; i++) {
    s = s + A[i];
}
prom = s / (double)n;
printf("Suma:%d\nProm:%d", s, prom);
```

3 Ejercicio 03

Ordenamiento de N números del 1 al 100.

```
for (int i=0; i<n-1; i++) {
    for (int k=i+1; k<n; k++) {
        if (vec[i] > vec[k]) {
            int aux = vec[i];
            vec[i] = vec[k];
            vec[k] = aux;
        }
    }
}
```

4 Ejercicio 04

Ordenamiento de N números del 1 al 100 (2.0).

```
int frec[101]={0};
for (int i=0; i<n; i++) {
    frec[vec[i]]++;
}
```

```
int pos = 0;
for (int i=0; i<101; i++) {
    for (int k=0; k<fec[i]; k++) {
        vec[pos] = i;
        pos++;
    }
}
```

5 Ejercicio 05

Algoritmo raro - Infinito.

```
int i=0;
int sum=0;
while (i < 100) {
    if (i % 2 == 0) {
        for (int k=0; k<n; k++) {
            sum += vec[i];
        }
    }
    else {
        for (int k=0; k<i; k++) {
            sum += vec[i];
        }
    }
}
```

6 Ejercicio 06

Factorial.

```
int fact = 1;
for (int i=2; i<n; i++) {
    fact *= i;
}
printf("Factorial: %d", fact);
```

7 Ejercicio 07

Buscar cadena de máximo 50 caracteres.

```
int pos = -1;
for (int i=0; i<n; i++) {
    if (strcmp(vec[i], cadBuscar) == 0) {
        pos = i;
        break;
    }
}
```

8 Ejercicio 08

Buscar el mayor.

```
int pos = 0;
for (int i=1; i<n; i++) {
    if (vec[i] > vec[pos])
```

```

    pos = i;
}
printf("El mayor es: %d", vec[pos]);

```

9 Ejercicio 09

Otro algoritmo raro – El while es válido?.

```

int max = 0;
for (int m=0; m<n; m++) {
    int cont = 0;
    int k = m + 1;
    while (vec[m] <= vec[k]) {
        k = k + 1;
        cont++;
    }
    if (cont > max)
        max = cont;
}
printf("Maximo %d", max);

```

10 Ejercicio 10

Logarítmica.

```

int i=1;
while (i < n) {
    if (vec[i] % 2 == 0)
        i *= 3;
    else
        i *= 2;
}

```

11 Ejercicio 11

Logarítmica (2.0).

```

int i=1;
while (i < n) {
    if (vec[i] % 2 == 0)
        n = n / 3;
    else
        n = n / 2;
}

```

12 Ejercicio 12

Búsqueda binaria – Arreglo ordenado.

```

int inf = 0;
int sup = n - 1;
int pos = -1;

while ((pos == -1) && (sup >= inf)) {
    int medio = (inf + sup) / 2;
    if (arreglo[medio] == 80)

```

```
    posicion = medio;
else if (arreglo[medio] < 80)
    limiteInferior = medio + 1;
else
    limiteSuperior = medio - 1;
}
```

13 Ejercicio 13-22

Implemente y analice las siguientes funciones.

1. Encontrar el número mayor en un arreglo de enteros
2. Ordenar un arreglo de números enteros
3. Eliminar el elemento en una posición de un arreglo
4. Buscar un número en un arreglo
5. Calcular el factorial de N
6. Determinar si un número existe en un arreglo de enteros.
7. Calcular cuántas veces se repite un número X en un arreglo de enteros.
8. Sumar los dígitos de un número entero positivo.
9. Determinar si un número es primo o no.
10. Determinar la cantidad de primos que existen en un arreglo de enteros.