

Programación y Algoritmos

Parte 1: Lenguaje C

Tarea #3

Fecha de entrega: 3/9/2021.

Problema 1. Programa que verifique si un número de entrada dado es palíndromo. Un **palíndromo**, es un número o string que se lee igual en un sentido que en otro (por ejemplo: Ana, 121...). Usar una función que regrese un número invertido (al revés).

Firma de la función: `int invert(int number)`

`Invert(123) →` regresa 321

Impresión en terminal: "El número ### es (o no es) palíndromo"

NOTA: No usar recursión ni arreglos.

Problema 2. Escribe un programa que implemente la estructura de datos Stack usando un arreglo de enteros. Se puede asumir que la capacidad máxima del Stack es de 10 enteros.

El arreglo se puede definir globalmente y se implementaran las funciones *push* y *pop*:

```
void push(int)
int pop( )
```

Simula 10 operaciones e imprime en terminal el estado interno del Stack.

Problema 3. Escribe una función que reciba un arreglo de enteros y que calcule el producto acumulado para cada entrada usando

$$a[i] = \prod_{j \neq i} a[j]$$

Se espera que la complejidad del algoritmo sea lineal $O(n)$. Se puede asumir que la longitud máxima del arreglo será 50.

Firma de la función: `products(int a[], int products[])`
Imprime en terminal el arreglo original y el arreglo de productos.

NOTA: El martes 31 de agosto estudiaremos funciones con argumentos de tipo arreglo.

Problema 4. Escribe una función que reciba dos arreglos de enteros ordenados en forma no decreciente, y combine dichos arreglos en un solo arreglo también ordenado en forma no decreciente.

Firma de la función:

```
void merge(int a1[], int n1, int a2[], int n2, int result[], int n3)
```

Se puede asumir que la longitud máxima de los arreglos a1 y a2 será 20, n1 y n2 representa el número de elementos en dichos arreglos.