



Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Ciencia de la Computación
Fundamentos de Programación

Primera Práctica Calificada Secciones A, B, C

CC112

11/04/2024 Tiempo: 2:30 horas

Ciclo: 2024-I

Normas:

1. No compartir respuestas/consultas con sus compañeros a través de chats, redes sociales u otros medios digitales.
2. No se permiten apuntes de clase.
3. Las soluciones serán enviadas a la plataforma y/o a la cuenta de correo del profesor.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos: _____ Nombres: _____

Código: _____ Sección: ____

1. [5.0 pts.] Una frase se considera palíndromo si invertida sin considerar los espacios es igual a la original sin considerar los espacios y sin tener en cuenta acentos. Los siguientes son ejemplos de frases palíndromas: "*dabale arroz a la zorra el abad*"; "*a ti no bonita*"; "*animo Romina*"; "*La ruta nos aporota otro paso natural*"; "*Se verla al revés*". Utilizando recursión escriba un programa en C++ que determine si una frase es o no un palíndromo.

Sugerencias:

- Para comprobar si un carácter es de tipo espacio, use: la función *isspace()*.
 - Para convertir a minúsculas use la función *tolower()*.
2. [5.0 pts.] A inicios de enero en una subasta de mercadería decomisada por la SUNAT se ofrece un lote de N panetones. El precio de cada panetón se almacena en un arreglo **PRECIO** y el mes de la fecha de vencimiento se almacena en un arreglo **MES**, considere que todos panetones vencen este año y que el precio es menor si la fecha de vencimiento es cercana. No compre panetones que se venzan en marzo y abril porque en esos meses el público no suele comprar panetones. Encuentre el precio mínimo a pagar por K panetones.

Por ejemplo, si

PRECIO = [60, 55, 30, 25, 45, 15, 30, 20, 50, 45]
MES = [10, 9, 4, 3, 7, 1, 4, 2, 8, 7]

Entonces el programa puede mostrar las siguientes salidas:

```

Numero de panetones a comprar: 11
Solo se pudieron comprar 10 panetones.
Monto gastado en la compra: 290 soles.
-----
Process exited after 62.04 seconds with return value 0
Presione una tecla para continuar . . .

```

```

Numero de panetones a comprar: 5
Monto gastado en la compra: 175 soles.
-----
Process exited after 3.16 seconds with return value 0
Presione una tecla para continuar . . .

```

3. [5.0 ptos.] Son años bisiestos todos los años múltiplos de 4, excepto aquellos que son múltiplos de 100 pero no de 400. Implementar un programa que reciba por teclado dos años, N año inferior y M año superior, y que almacene en un arreglo (array) todos los años bisiestos comprendidos entre ambos años. Primero se debe calcular la cantidad de bisiestos y luego almacenarlos. Luego el programa debe pedir que se ingrese un año por teclado y utilizando búsqueda lineal determinar el año bisiesto inmediato inferior, de lo contrario muestre un mensaje “año fuera de rango”.

Ejemplos de salidas.

Ingrese año inicial: 2000

Ingrese año final: 2100

la lista de bisiestos entre 2000 y 2100 es

**2000 2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060 2064
2068 2072 2076 2080 2084 2088 2092 2096**

Ingrese un año a buscar: 2018

el año bisiesto inmediato inferior es: 2016

Ingrese año inicial: 1900

Ingrese año final: 2000

la lista de bisiestos entre 1900 y 2000 es

**1904 1908 1912 1916 1920 1924 1928 1932 1936 1940 1944 1948 1952 1956 1960 1964 1968
1972 1976 1980 1984 1988 1992 1996 2000**

Ingrese un año a buscar: 1924

el año bisiesto inmediato inferior es: 1920

Ingrese año inicial: 1800

Ingrese año final: 1900

la lista de bisiestos entre 1800 y 1900 es

**1804 1808 1812 1816 1820 1824 1828 1832 1836 1840 1844 1848 1852 1856 1860 1864 1868
1872 1876 1880 1884 1888 1892 1896**

Ingrese un año a buscar: 2010

año fuera del rango

4. [5.0 ptos.] Dado un arreglo de enteros A de tamaño n , ordenado de manera ascendente, cuyos elementos son todos distintos, implemente las siguientes funciones:

- Función que permita desplazar los elementos del arreglo con índice $1 \leq k < n$, de manera que

el arreglo resultante sea:

$[A[k], A[k + 1], \dots, A[n - 1], A[0], A[1], \dots, A[k - 1]]$

Ejemplo:

Si $A = [0, 1, 2, 3, 4, 5, 6, 7]$, con $k=3$, se obtiene $[4, 5, 6, 7, 0, 1, 2]$

- Implemente otra función que tome como argumento el arreglo desplazado en a), su tamaño y un entero c y retorne el índice de c , si c está en el arreglo desplazado, o -1 en caso contrario. Su implementación debe tener complejidad temporal $O(\log n)$.

Ejemplos:

Input: $[4, 5, 6, 7, 0, 1, 2]$, $c=0$

Output: 1

Input: $[4, 5, 6, 7, 0, 1, 2]$, $c=8$

Output: -1

Nota: Cuando el tamaño de entrada disminuye en cada iteración o paso, se dice que un algoritmo tiene complejidad logarítmica. Este método es el segundo mejor porque su programa se ejecuta con la mitad del tamaño de entrada en lugar del tamaño completo.