



CC102

Segunda Práctica Calificada
16/9/2019 Tiempo: 2 horas

Ciclo: 2019-2

Normas:

1. El alumno entregará esta hoja de examen debidamente llenada con sus datos.
2. La solución de la prueba se guardarán en **Escritorio**, carpeta: **ApellidoNombreCodigo**(sin espacios en blanco), la pregunta **n** se guardará en el archivo: **n.c** (n = 1, 2, ..).
3. No se permite: El uso de celulares, internet, USB, ingresar después de 15 min. de iniciado el examen ni salir antes de la hora de finalización.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos : _____ Nombres : _____

Sección : _____ Grupo: _____

1. [5 ptos.] Se sabe que los números primos menores que 100 son 25. Defina el vector **primos** donde se debe guardar en forma creciente todos los primos menores que 100. Finalmente, imprima dicho vector.
2. [5 ptos.] Continuando con la pregunta anterior, luego de generar el vector **primos** y antes de imprimir dicho vector, implemente (complete):

```
for (int j = 25; j > 1; j--)  
{  
    k = rand() % j;  
    // intercambiar los valores de primos[j - 1] y primos[k]  
}
```

Finalmente, imprima un mensaje explicando lo que hacen las instrucciones del script, mostrado líneas arriba.

3. [4 ptos.] Continuando con la pregunta anterior, ordene el vector **primos** según el algoritmo de ordenamiento de la burbuja (bubble sort). Finalmente, imprima el número de comparaciones que se realizó para dicho ordenamiento.
4. [6 ptos.] Repita lo realizado en la pregunta anterior, solo que esta vez debe ordenar el vector **primos** usando alguno de los algoritmos de ordenación vistos en clase (**quick sort**, **selection sort** o **insertion sort**). Finalmente, imprima el número de comparaciones que se realizó para dicho ordenamiento y vuelva a imprimir dicho vector.

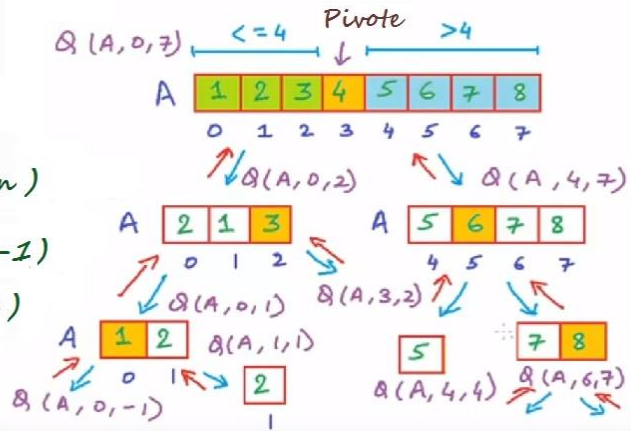
QUICK SORT (ordenación rápida)

QuickSort(A, inicio, fin)

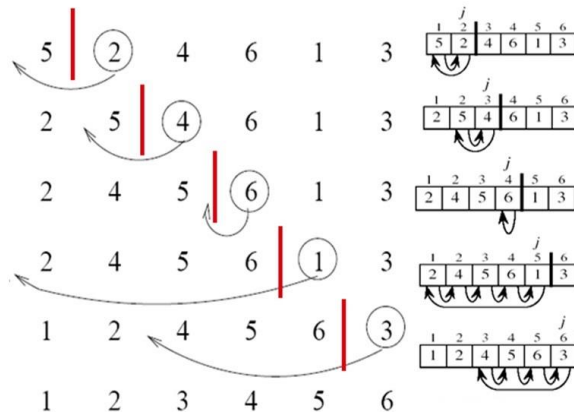
```

{
  if ( inicio < fin )
  {
    pIndice ← Particion (A, inicio, fin)
    QuickSort(A, inicio, pIndice - 1)
    QuickSort(A, pIndice + 1, fin)
  }
}

```



INSERTION SORT (ordenación por inserción)



SELECTION SORT (ordenación por selección)

