

Auxiliatura INF-131 “C”

Estructura de Datos y Algoritmos

Univ. Miguel Angel Quispe Mamani

Universidad Mayor de San Andrés

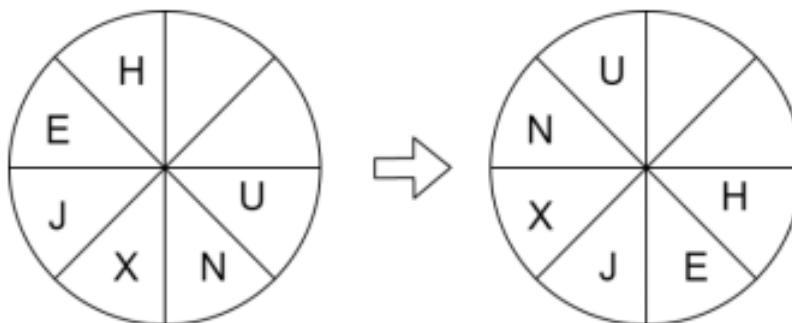
Carrera de Informática

27/02/2023

Ejercicio 1

Sea una cola circular de caracteres se pide:

1. Mover los caracteres que más se repiten al final de la cola.
2. Invertir el orden de la cola sin el uso de estructuras auxiliares



```
1 cantidadDeOcurrencias(ColaCircularC w, char x) {  
2     cont = 0  
3     n = w.nroelem()  
4     for i = 0 to n - 1 {  
5         elem = w.eliminar()  
6         if elem = x  
7             cont = cont + 1  
8         w.adicionar(elem)  
9     }  
10    return cont  
11 }
```

```
1 maximoRepetido(ColaCircularC w) {
2     maximo = 0
3     n = w.nroelem()
4     for i = 0 to n - 1 {
5         elem = w.eliminar()
6         w.adicionar(elem)
7         ans = cantidadDeOcurrencias(w, elem)
8         if ans > maximo
9             maximo = ans
10    }
11    return maximo
12 }
```

```
1 llevarAlFinal(ColaCircularC w) {
2     int nro = w.nroelem()
3     for i = 0 to nro - 1 {
4         maximoRepe = maximoRepetido(w)
5         a, b = new ColaCircularC()
6         while not w.esvacía() {
7             elem = w.eliminar()
8             cantidad = cantidadDeOcurrencias(w, elem) + 1
9             if cantidad = maximoRepe {
10                a.adicionar(elem)
11                while not w.esvacía() {
12                    elem2 = w.eliminar()
13                    if elem2 = elem
14                        a.adicionar(elem2)
15                    else
16                        b.adicionar(elem2)
17                }
18            } else
19                b.adicionar(elem)
20        }
21        w.vaciar(b)
22        w.vaciar(a)
23    }
24 }
```

```
1 invertir1(ColaCircularC w){
2     a = new ColaCircularC()
3     nro = w.nroelem()
4     for r = 1 to nro{
5         n = w.nroelem()
6         for i = 1 to n - 1
7             w.adicionar(w.eliminar())
8             a.adicionar(w.eliminar())
9         }
10        w.vaciar(a)
11    }
```

```
1 invertir2(ColaCircularC w){
2     n = w.nroelem()
3     for i = 0 to n - 1{
4         for j = 1 to n - i - 1
5             w.adicionar(w.eliminar())
6         x = w.eliminar()
7         for j = 0 to i - 1
8             w.adicionar(w.eliminar())
9         w.adicionar(x)
10    }
11 }
```

```
1 Begin
2     a = new ColaCircularC()
3     read(n)
4     a.llenar(n)
5     a.mostrar()
6     //Inciso 1
7     llevarAlFinal(a)
8     a.mostrar()
9     //Inciso 2
10    invertir1(a) // con una estructura auxiliar
11    invertir2(a) // sin estructuras auxiliares
12    a.mostrar()
13 End
```

Ejercicio 2

Una cafetería utiliza las siguientes estructuras para organizar su información:

- Cola circular de productos <idProd, nombre, precio>
- Pila de clientes <ci, nombre>
- Cola de Ventas <ci, idProd, cantidad, fecha>

1. Mostrar el(los) producto(s) más vendido(s).
2. Ordenar a los clientes descendientemente por el monto total gastado.

```
1 maximoProdVendido(ColaCircularP a, CSimpleV b){
2     maximo = 0
3     nroProd = a.nroelem()
4     for i to nroProd{
5         x = a.eliminar()//Producto
6         a.adicionar(x)
7         nroVent = b.nroelem()
8         cont = 0
9         for j = 1 to nroVent{
10            y = b.eliminar()//Venta
11            b.adicionar(y)
12            if x.getIdProd() == y.getIdProd()
13                cont = cont + 1
14        }
15        if cont > maximo
16            maximo = cont
17    }
18    return maximo
19 }
```

```
1 productosMasVendidos(ColaCircularP a, CSimpleV b){
2     maximo = maximoProdVendido(a, b)
3     nroProd = a.nroelem()
4     for i to nroProd{
5         x = a.eliminar()//Producto
6         a.adicionar(x)
7         nroVent = b.nroelem()
8         cont = 0
9         for j = 1 to nroVent{
10            y = b.eliminar()//Venta
11            b.adicionar(y)
12            if x.getIdProd() == y.getIdProd()
13                cont = cont + 1
14        }
15        if cont = maximo
16            x.mostrar()
17    }
18 }
```

```

1 gastoTotal(ColaCircularP a, CSimpleV b, string ci){
2     nroVentas = b.nroelem()
3     costo = 0
4     for i = 1 to nroVentas{
5         x = b.eliminar()
6         b.adicionar(x)
7         if b.getCi() == ci{
8             nroProductos = a.nroelem()
9             for j = 1 to nroProductos{
10                y = a.eliminar()
11                a.adicionar(x)
12                if y.getIdProd() == x.getIdProd()
13                    costo = costo + (y.getPrecio() * x.getCantidad())
14            }
15        }
16    }
17    return costo
18 }

```

```

1 ordenar(ColaCircularP a, CSimpleV b, PilaCl c){
2     u, v = new PilaCl()
3     elem = c.eliminar()
4     while not c.esvacía(){
5         x = c.eliminar()//Cliente
6         while not c.esvacía(){
7             if gastoTotal(a, b, x.getCi()) > gastoTotal(a, b, elem.getCi()){
8                 u.adicionar(elem)
9                 elem = x
10            }else{
11                u.adicionar(x)
12            }
13        }
14        c.vaciar(u)
15        v.adicionar(elem)//aca esta el mayor
16    }
17    c.vaciar(v)
18 }

```

```

1 Begin
2     a = new ColaCircularP()//Cola de Productos
3     b = new CSimpleV()//Cola de Ventas
4     c = new PilaCl()//Pila de Clientes
5     a.llenar(5)
6     b.llenar(10)
7     c.llenar(4)
8     a.mostrar()
9     b.mostrar()
10    c.mostrar()
11    //Inciso 1
12    productosMasVendidos(a, b)
13    //Inciso 2
14    ordenar(a, b, c)
15    c.mostrar()
16 End

```