

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Primer Examen

(Primer Semestre 2022)

Duración: 2h 50 min.

- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Los programas deben ser desarrollados en el lenguaje C. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI. Tampoco está permitido usar código obtenido en internet.
- Para este examen solo se permite el uso de las librerías **stdio.h**, **stdlib.h** y **math.h**
- **No están permitidas las funciones para obtener el tamaño de la pila o cola que recorran estas estructuras.**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo_EX1_PY**

Resuelva solo 2 de las siguientes preguntas:

Pregunta 1 (10 puntos)

Acaba de llegar un contenedor con n productos alimenticios que luego deberán ser guardados, por un robot, en p cajas pequeñas de capacidad c productos con la consideración de que los que tienen fecha de caducidad posterior siempre irán por debajo de los que tienen fecha de caducidad anterior. Se deduce que el redondeo hacia arriba de (n/c) es igual a p . El contenedor y las cajas pequeñas se comportan como pilas y sólo se pueden utilizar estas pilas para lograr la tarea de guardar los productos. Al final, se tendrán que embalar las cajas con los productos cuyas fechas de caducidad están más alejadas a la fecha actual y dejar abierta la caja que contiene a los productos que están próximos a vencer. **Recuerde que la pila contenedor es la única que puede guardar los productos sin importar el orden por la fecha de caducidad. También recuerde que no se puede superar la capacidad de cada pila caja.**

Si por ejemplo el contenedor viene con $n=10$ productos en el orden cómo se presenta en la figura 1 y la capacidad de cada caja es de $c=3$, entonces se tendrán 4 cajas en las que se tienen que guardar los productos siguiendo las consideraciones previamente descritas. Luego de las operaciones de movimientos de productos para guardarlos en las cajas se tendrá la distribución de los productos en las 4 cajas, tal como se muestra en la figura 2. Puede definir un arreglo de cajas para su almacenamiento en memoria.

Id del producto Fecha de caducidad	10 20221224	9 20221130	8 20221212	7 20220814	6 20221106	5 20221003	4 20220918	3 20220630	2 20221006	1 20110603

Figura 1. Pila del contenedor de productos.

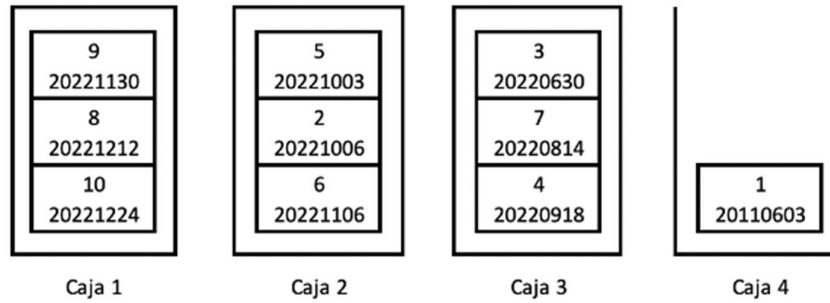


Figura 2. Resultado final luego de la distribución de los productos en las 4 cajas.

A continuación, les mostramos un ejemplo del ingreso de datos que requiere este programa:

Cantidad de productos (n): 10

Capacidad de cada caja (c): 3

Código de producto	10	9	8	7	6	5	4	3	2	1
Fecha de caducidad	20221224	20221130	20221212	20220814	20221106	20221003	20220918	20220630	20221006	20110603

Con los datos brindados el resultado sería:

Caja 1: 9, 8, 10

Caja 2: 5, 2, 6

Caja 3: 3, 7, 4

Caja 4: 1

Se le solicita implementar lo siguiente:

- Las estructuras de datos que permitan soportar lo descrito. (1.0 punto)
- Una función o funciones que permitan distribuir los productos de la pila contenedor a las pilas cajas de acuerdo con las consideraciones descritas. Para la distribución de los productos a las pilas cajas puede usar el contenedor pila inclusive. Sólo puede usar las operaciones de pila (*apilar*, *desapilar*, *cima* y *es_pila_vacia*). **No pueden usar otras estructuras como arreglos, listas, colas u otras pilas para resolver la distribución de los productos, sólo las pilas del contenedor y las cajas.** Sí puede guardar las pilas correspondientes a las cajas en un arreglo. Puede usar iteraciones o recursividad. (7.5 puntos)
- Las funciones necesarias incluyendo la principal para realizar la simulación descrita. La entrada de datos se puede hacer mediante archivo, consola o un arreglo con los datos que luego deberán ser colocados en el contenedor pila. (1.5 puntos)

Pregunta 2 (10 puntos)

Un conocido Banco del Perú (BRC) ha decidido adquirir un robot para la entrega de tickets de atención para sus agencias. Luego de un tiempo de uso, los clientes empiezan a quejarse ya que el robot se demora mucho en la generación de los tickets. Por tal motivo se le solicita que revise los algoritmos que usa en su programación. Luego de revisarlo se da cuenta que el algoritmo que utiliza el robot para el manejo de colas con prioridad tiene una complejidad de $O(n^2)$ lo que debe estar causando la demora, especialmente cuando se tiene gran cantidad de clientes en

espera. Se sabe que el banco tiene 3 tipos de clientes: vip(V), cliente(C) y no cliente (N), brindando prioridad en atención a los vips sobre los clientes y no clientes, de la misma forma priorizando a los clientes sobre los no clientes.

Posteriormente a la entrega de su informe se le da la indicación de que debe mejorar los algoritmos que usa el robot, así que, revisando el código fuente se da con una gran sorpresa, que en el código aparece como autor "Andres M." así que recuerda que su profesor manejaba las colas mediante listas simplemente ligadas para todas las operaciones, por lo cual es posible que dentro de estas bibliotecas se realicen recorridos sobre las listas u ordenamientos que demandan mucho tiempo. Por tal motivo Ud. debe adaptar estas bibliotecas para que la complejidad de la generación o atención de los tickets tenga una complejidad constante $O(1)$ a todo nivel, sea en bibliotecas de la cola como en el de las listas (solo puede tener una complejidad mayor el proceso de impresión de la lista). **Para esta tarea es posible que pueda acceder directamente a elementos de la cola, pero sin recorrerla, así como realizar modificaciones en las estructuras de la lista o de la cola.**

A continuación, se aprecian algunos ingresos de datos y sus salidas respectivas:

Ingresos de clientes a la cola de atención:

('V',1) ('N',1) ('N',2) ('V',2) ('C',1) ('C',2) ('C',3) ('N',3) ('V',3)

La cola se muestra así:

[V 1, V 2, V 3, C 1, C 2, C 3, N 1, N 2, N 3]

Si luego se atienden a dos clientes, la cola quedaría así:

[V 3, C 1, C 2, C 3, N 1, N 2, N 3]

- a) Defina las estructuras necesarias para soportar lo solicitado, además, por temas de hardware, el robot siempre maneja nodos con un **carácter** que indica el tipo de cliente y un número **entero** que indica su orden dentro del grupo al que pertenece. **Si usa otra representación la siguiente pregunta no tiene validez** (1.0 punto)
- b) Adapte las bibliotecas de listas y colas, para manejar una cola con prioridad que tenga la complejidad solicitada de acuerdo a las restricciones indicadas. **Recuerde que solo puede usar una cola para esta tarea, cualquier otra estructura auxiliar no está permitida** (9.0 puntos)

Pregunta 3 (10 puntos)

Un robot utiliza *round robin* para asignar intervalos de tiempo de igual longitud de ejecución en minutos (T) a los diferentes procesos que el usuario le asigna. Estos procesos se ejecutan de manera secuencial, uno después del otro, y solo durante el tiempo asignado. Si un proceso ha concluido, se remueve de la lista de procesos pendientes, y se continúa el ciclo hasta que no queden más procesos por ejecutar. Un proceso que no termina durante un intervalo de tiempo asignado, se pone en pausa, y continuará la siguiente vez que el sistema operativo le asigne recursos. Finalmente, si el usuario solicita la ejecución de un nuevo proceso, este se agregará inmediatamente después del proceso actual en ejecución.

Se desea diseñar un programa sencillo utilizando una única lista, que simule la solicitud de ejecución, y la asignación de recursos por parte del robot para diferentes procesos. Los procesos y sus tiempos de ejecución serán ingresados desde el terminal, como se puede apreciar en el siguiente ejemplo. Tenga en cuenta que colocar 1 cuando se pregunta si se desea insertar un nuevo proceso corresponde a **si**, mientras que colocar 0 corresponde a **no**.

Nota: no se pueden utilizar estructuras o espacio de memoria adicionales a la lista indicada.

procesos iniciales = 3, T = 1 min

Duraciones iniciales de procesos (en minutos, respectivamente) = 2, 3, 2

Se ejecutó proceso #1

Desea insertar un nuevo proceso? 1

Indicar duración de proceso #4: 2

Se ejecutó proceso #4

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #2

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #3

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #1

Se completó proceso #1

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #4

Se completó proceso #4

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #2

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #3

Se completó proceso #3

Desea insertar un nuevo proceso? 0

Se ejecutó proceso #2

Se completó proceso #2

Desea insertar un nuevo proceso? 0

a) Implementar el ingreso de datos por teclado para los datos (1p)

b) Desarrollar las estructuras necesarias para poder realizar la simulación requerida (9p)

Profesores del curso:

Johan Baldeon

Rony Cueva

Juan M. Chau

San Miguel, 21 de mayo del 2022