

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

6ta práctica (tipo b)
Segundo Semestre 2022

Indicaciones Generales:

- Duración: 110 minutos.

• SOLO ESTÁ PERMITIDO EL USO DE APUNTES DE CLASE. NO PUEDE UTILIZAR FOTOCOPIAS NI MATERIAL IMPRESO, TAMPOCO PODRÁ EMPLEAR HOJAS SUELTAS.

- No se pueden emplear variables globales, estructuras, ni objetos (con excepción de los elementos de iostream, iomanip y fstream). No puede utilizar la clase (o el tipo de datos) string. Tampoco se podrán emplear las funciones malloc, realloc, strdup o strtok, igualmente no se puede emplear cualquier función contenida en las bibliotecas stdio.h, cstdio o similares y que puedan estar también definidas en otras bibliotecas. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE LABORATORIO**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada función NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo main.cpp solo podrá contener la función main de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. En el archivo main.cpp deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontará 0.5 puntos en la nota final.
- El código comentado NO SE CALIFICARÁ. De igual manera NO SE CALIFICARÁ el código de una función si esta función no es llamada en ninguna parte del proyecto o su llamado está comentado.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.

SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO.

NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES DADAS EN LA PRUEBA

- **Puntaje total:** 20 puntos.

Cuestionario:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 5 del curso. En este laboratorio se trabajará con punteros sin tipo, así como la implementación de TAD's.

Deberá elaborar un proyecto denominado "LAB06_TrasladaPila" y en él desarrollará el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.

Se tienen los archivos que se describen a continuación:

Numeros.txt
13
4
1
9
7
...

Medicinas.csv
60509,AMPICILINA 125MG SUSP 90ML,58.65
73972,VITAMINA E 400 MG C90 CAPS,54.4
94429,TIAMAZOL 5MG 20T,52.7
...
Código,Nombre del medicamento,Peso de la caja de medicinas.

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```

#include "BibliotecaPilaGenerica.h"
#include "PilaConEnteros.h"
#include "PilaConRegistros.h"

using namespace std;

int main(int argc, char** argv) {
    void *arreglo, *pilaini, *pilafin;

    cargaarreglo(arreglo, cmpnumero, leenumero, "numeros.txt");
    cargapila(pilaini, arreglo);
    muevepila(pilaini, pilafin);
    imprimepila(pilafin, imprimenumero, "repnumeros.txt");

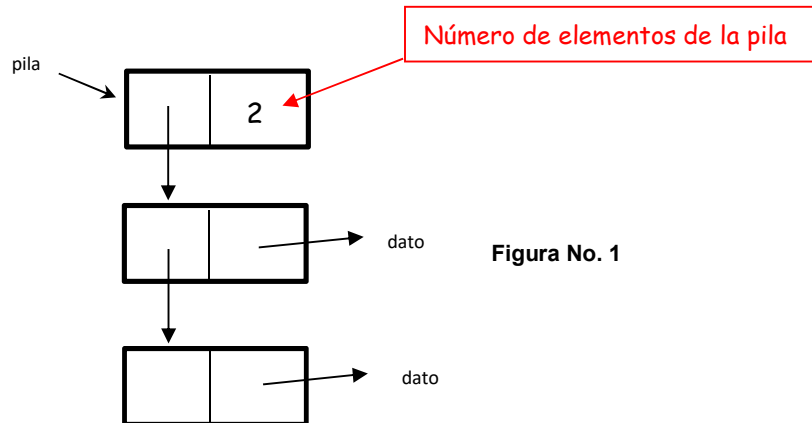
    cargaarreglo(arreglo, cmpregistro, leeregistro, "medicinas.csv");
    cargapila(pilaini, arreglo);
    muevepila(pilaini, pilafin);
    imprimepila(pilafin, imprimeregistro, "repmedicinas.txt");

    return 0;
}

```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

Se solicita transportar de una pila a otra los elementos contenidos, en nuestro caso representadas por las pilas genéricas **pilaini** y **pilafin**, considerando que en ningún momento se puede colocar un producto de mayor peso sobre alguno de menor peso, ya que los medicamentos se aplastarían. Las pilas genéricas tendrán la siguiente representación:



Esta estructura debe contar con una biblioteca **BibliotecaPilaGenerica**, con las funciones necesarias para su soporte, como son las funciones **cargapila**, **push**, **pop**, **pilavacia**, **imprimepila** y **muevepila**. Debido a que ordenar una pila es una operación compleja, se cuenta con un **arreglo genérico**, que utilizando la función **cargaarreglo** reciba el nombre del archivo a ser cargado y luego ordene sus elementos con el fin de ser ingresados a la pila **pilaini**, para el proceso de ordenamiento debe usar la función **qsort** de la biblioteca **stdlib**. La función **muevepila** debe mover los elementos de **pilaini** a **pilafin**, durante este proceso no puede colocar un peso mayor sobre un peso menor, **pero si puede usar una pila auxiliar. Recuerde que la pila no puede recorrerse**. La función **imprimepila** es la única que puede recorrer la pila pero solo para imprimir sus resultados.

Pregunta 1 (12 puntos)

Desarrolle la biblioteca **BibliotecaPilaGenerica**, que brinde soporte a las pilas que aparecen en el main de acuerdo con la figura 1. Para probar el buen funcionamiento de esta biblioteca genérica, desarrolle la biblioteca **PilaConEnteros** que trabaja **solo** con números enteros. Los números deben ingresarse en **pilaini** de tal forma que no exista un número con menor valor debajo de uno de mayor valor, para esta tarea debe

ordenar primeramente los valores en un arreglo genérico, empleando **qsort** mediante la función **cmpnumero**. Una vez que **pilaini** este cargada en orden, se deben pasar los números a **pilafin**, pero recuerde en ninguna de las operaciones de traslado debe colocar un valor mayor sobre otro menor. Para imprimir la pila, debe usar la función **imprimepila** empleando **imprimenúmero** que mostrará el contenido de la pila, de la siguiente forma:

```
1
4
5
6
...
```

Pregunta 2 (8 puntos)

Una vez compruebe el buen funcionamiento de su biblioteca genérica, desarrolle la biblioteca **PilaConRegistros**, donde deberá cargar todos los datos del archivo **Medicinas.csv** en un arreglo genérico, para luego ordenarlo empleando **qsort** utilizando la función **cmpregistro** de acuerdo al peso de cada producto, considerando no aplastar los productos con menos peso.

Luego de cargar **pilaini**, debe pasar los productos a **pilafin**, de acuerdo con las consideraciones de peso indicadas. Finalmente debe usar la función **imprimepila**, empleando **imprimeregistro** que deberá recorrer la pila e imprimir un reporte similar al que se muestra a continuación:

repmedicinas		
Codigo	Descripcion del Producto	Peso Caja
11370	KETOROLACO 4T 30MG SUBLINGUAL	2.55
16813	VANCOMICINA 500MG SOL INY	5.95
33311	ALBENDAZOL 200MG C/6 TAB	20.40
26484	DILTIAZEM 30 TAB/30 MG	29.75
81333	CARNITINA C/30	40.80
62302	BEZAFIBRATO 200MG C/30TAB	41.65
...		

Un algoritmo muy conocido para realizar la operación de mover elementos de una pila a otra sin aplastar los menos pesados es el algoritmo Hanoi, a continuación, se muestra su pseudocódigo:

Hanoi(n, origen, auxiliar, destino)

Si n == 1 entonces

Sacar disco de la torre origen.

insertar disco a torre destino.

Sino

Hanoi(n-1, origen, destino, auxiliar) //mover todas los discos menos el más grande (n) a la torre auxiliar.

mover disco n a torre destino

Hanoi(n-1, auxiliar, origen, destino) //mover todos los discos restantes, encima del disco n.

FinSi

Fin

Al finalizar la práctica, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares.**

Profesores del curso: Miguel Guanira
Rony Cueva

San Miguel, 28 de octubre del 2022.