



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Apellido y Nombre:

DNI:

Calificación:

Tópicos de programación

03635

Final

06/11/2024

Parcial Tópicos de Programación

ATENCIÓN: para que el examen pueda ser corregido y archivado debe ser entregado obligatoriamente antes de la hora límite de las 18:00 teniendo en cuenta un horario de inicio 15:00 y con el siguiente formato: DNI_APELLIDO_NOMBRE.zip, ejemplo 41127133_PEREZ_MARIA_PIA_P.zip. Observe que es un .zip, no es .rar ni .7z respete la forma de entrega. La P como sufijo indica que es el PARCIAL. Adjunte el enunciado y elimine las carpetas *bin* y *obj*, está compartiendo programas y por obvias razones los servidores eliminarán comprimidos que contengan binarios o ejecutables.

Se adjunta un proyecto C en code::blocks. El proyecto administra una colección de movimientos de stock utilizando una biblioteca genérica denominada TDASimpleVec. En el ejemplo puntual, dada una colección de movimientos de stock se procede primero a ordenar el vector para luego obtener el acumulado de movimientos de stock por producto. Ejemplo:

PR01	10	PR02	5	PR01	-1	PR03	98	PR01	-5	PR02	2
------	----	------	---	------	----	------	----	------	----	------	---

Luego del **ordenamiento** por código de producto podría quedar de la siguiente manera:

PR01	10	PR01	-1	PR01	-5	PR02	5	PR02	2	PR03	98
------	----	------	----	------	----	------	---	------	---	------	----


Y luego en un único recorrido se deben obtener los totales del **agrupamiento** de movimientos por producto, transformándose en la siguiente colección:

PR01	4	PR02	7	PR03	98
------	---	------	---	------	----

Se aclara que todas las operaciones deben ser realizadas sobre el mismo espacio de memoria. Quite el comentario a la macro #define EXAMEN e implemente usted las funciones de creación, ordenamiento, agrupamiento y destrucción del TDA. Puede utilizar todas las funciones de biblioteca ANSI C que necesite. Las funciones deberán respetar el prototipo entregado, la macro solo modifica el nombre de la función agregando el sufijo ALU para que usted pueda implementar de a partes. **De ninguna manera modifique los prototipos de las funciones entregadas, es condición necesaria para la aprobación que esto se mantenga.** Recuerde que la prueba entregada es solo un ejemplo, usted es responsable de probar y controlar que las implementaciones funcionan en todos los casos. Las implementaciones se tendrán que mantener genéricas, si quiero implementar la misma idea para sumar la cantidad de kilómetros de una serie de vehículos en lugar de acumular stocks, su implementación debe funcionar de igual forma.

Condiciones para la aprobación:

- La biblioteca debe poder almacenar tantos elementos como la memoria total disponible en el equipo lo permita (No debe estar limitado a la memoria de pila).
- El TDASimpleVec tiene que permitir administrar colecciones de cualquier tipo, usted debe mantenerlo genérico.
- La operación de agrupamiento con el vector ordenado se resuelve con una única recorrida.
- Vectores y cadenas de texto deberán ser manipulados utilizando aritmética de punteros.
- Las soluciones tienen que ser eficientes:
 - En el uso de memoria, por tanto, no declare vectores o matrices auxiliares si no es necesario

 UNLaM Dto. Ingeniería e Investigaciones Tecnológicas	Tópicos de programación 03635 Final 06/11/2024
Apellido y Nombre:	
DNI:	
Calificación:	

- **En cantidad de ciclos de procesador y en el caso de matrices las soluciones deben ser óptimas.**
- **No acceda nunca a memoria que no le pertenece y nunca deje memoria sin liberar.**
- **Declare variables al inicio del bloque y no utilice VLA (Variable length arrays)**
- **Se considerará también el uso de nombre descriptivos, separación de funciones y prolijidad general.**