



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

(0612) PROGRAMACIÓN II
(1110) PROGRAMACIÓN
FINAL MARZO
MESA B
19 hs
01/03/2023

Apellido y Nombre:

DNI:

Calificación :

Ejercicio C

El departamento de compras de la empresa “Mi Empresa”, cada mes procesa un archivo de texto “pedidos.txt” donde cada línea representa un pedido de un producto. Es un archivo de texto de longitud variable con ‘|’ como separador de campos. Este archivo no tiene ningún tipo de orden:

“nroPedido”: número de pedido (entero);

“codPro”: código del producto (alfanumérico de 10 caracteres como máximo);

“cantPedida”: cantidad de unidades del producto solicitadas (entero);

“codSector”: código del sector que realizó el pedido (1 carácter);

“oficina”: oficina que realiza el pedido (alfanumérico de 20 caracteres);

Se pide:

- crear una lista de **compras**: ordenada por código de producto, que acumula la cantidad solicitada total.
- Muestre la lista creada en el punto A).
- A partir de la lista creada en el punto A) genere y muestre por pantalla, un archivo binario denominado “resultadoCompras.bin” ordenado por cantidad solicitada total.

Debe crear un lote de prueba.

Trabaje de forma modular y lo más genérico posible.

Entregar con con 0 errores, 0 warnings y funcionando correctamente.

Ejercicio C++

Desarrolle la clase “Medicion” para que el código “main” provisto al pie sea válido. Los objetos de la clase “Medicion” almacenan el valor de una medición en flotante y su unidad de medida como cadena de caracteres. Las mediciones se pueden operar siempre y cuando su unidad de medida sea la misma, caso contrario la operación no es permitida. También es posible, como se indica en el código al pie sumar a un flotante la “Medicion”. En este caso no es importante la unidad de medida ya que existe una sola. Use efectivamente el tiempo y no desarrolle nada que no aplique al código mostrado.

Si el código entregado contiene errores comente la línea indicando cual es el error y luego escriba la línea en forma correcta.

```
Medicion m1mv(100.0, "Mv");//millivolts
Medicion m2mv(20.0, "Mv");
Medicion m4amp(3.0, "Amp");
Medicion m3mv=m1mv-m2mv;
cout<<"Resultado 1: "<<180.0+m3mv<<endl;

try{
    cout<<m3mv-m4amp<<endl;
}catch(...){
    cout<<"No se pueden restar mediciones de distinta unidad de medida"<<endl;
}
```

Crear la clase solicitada desarrollando lo mínimo indispensable para que la misma funcione correctamente. No olvide poner atención en el uso de los “const”.
Entregar con 0 errores, 0 warnings y funcionando correctamente.

EVALUACIÓN

- Desarrolle cada ejercicio en un proyecto separado.
 - **Incluya en el encabezado de cada archivo, // apellido_nombre_DNI**
 - Recuerde antes de comprimir, eliminar las carpetas bin y obj de cada proyecto.
 - **Entregue ambos proyectos compactados en un zip, “apellido_nombre_DNI.zip”.**
 - AMBOS EJERCICIOS: deben entregarse con 0 errores, 0 warnings y funcionando correctamente.
 - Entregue el parcial usando prácticas de MIEL.
-
- ¡La evaluación es individual!

¡El mayor de los éxitos!