

ITCR – ANÁLISIS Y DISEÑO DE ALGORITMOS

Tarea 3: uso del componente <vector> de C++

INSTRUCCIONES:

- Nombre del proyecto en C++: **tarea3_su_nombre**
- Enviar un archivo comprimido (solo .zip) con todos los objetos necesarios para ejecutar el programa.
- Recuerde usar buenas prácticas de programación como nombres significativos, reutilización de código (funciones), documentación interna, eficiencia, etc.
- No valide datos, estos se darán según las especificaciones.
- Enviar al tecDigital / EVALUACIONES / TAREAS. Verifique el envío al tecDigital.
- Requisito para calificar la tarea:
 - Tener documentación interna, al menos en cada función debe poner comentarios de lo que hace la función, sus entradas y salidas.
- Fecha de entrega: 15 de setiembre, 11pm

RECOMENDACIONES:

- Lea cuidadosamente el problema a resolver (primer paso de la metodología de solución de problemas: entender el problema).
- En la etapa de diseño del algoritmo se sugiere hacer un esquema donde determine el comportamiento del algoritmo para luego proceder con su desarrollo en la computadora.

Objetivos de la tarea:

+ Manipular la estructura de datos ofrecida por la plantilla de clase <vector> de C++:

- Insertar elementos.
- Modificar elementos.
- Eliminar elementos.
- Consultar elementos.

+ Uso de struct para definir la estructura de cada elemento del vector.

La combinación de struct (o class) con vectores o arreglos nos permiten tener en ellos datos de tipos diferentes.

Hay que desarrollar un programa para que la Fábrica de Arduinos pueda controlar la producción de placas de arduinos. La estructura de datos, usando struct, donde están las cantidades a producir es un vector cuyos elementos tienen esta forma:

tipo de arduino (string)
cantidad de arduinos de ese tipo que se van a producir en el mes
(entero ≥ 1)

En C++:

```
struct produccion
{
    string arduino;
    int cantidad;
};
```

Ejemplo de un posible vector de producción:

```
vector <produccion> vector_produccion;
```

Note que los elementos del vector son del tipo **produccion**: la estructura definida anteriormente. No se pone la cantidad de elementos en el vector, es dinámico.

Datos de ejemplo para el vector vector_produccion:

Elemento 1: {"Arduino UNO", 2000}

Elemento 2: {"Arduino Zero Pro", 3500}

Elemento 3: {"Arduino BT", 3000}

Elemento 4: {"Arduino Industrial 101", 1000}

Note que los elementos del vector son del tipo de estructura definida anteriormente.

De este vector podemos determinar que del tipo "Arduino UNO" se deben producir 2000 unidades, del tipo "Arduino Zero Pro" se deben producir 3500 unidades, etc.

Adicionalmente al vector de producción, la fábrica tiene una lista de los tipos de arduinos que puede producir. Estos tipos están en un vector (nativo o integrado de C++) definido como constante. Haga esta definición después de los #include.

```
const string tipos_de_arduino[] = {  
    "UNO",  
    "TRE",  
    "Zero",  
    "Zero Pro",  
    "BT",  
    "Mega",  
    "Ethernet",  
    "Pro",  
    "Pro Mini",  
    "Micro",  
    "Primo",  
    "Nano",  
    "Industrial 101",  
    "LilyPad",  
    "Esplora" };
```

Aunque la fábrica puede producir cualquier tipo de arduino que se encuentre en esa lista, en el vector de producción solamente deben estar los tipos que tengan unidades a producir. Los tipos de arduinos que tengan 0 en producción no deben estar en dicho vector.

Para manejar el vector se requieren estas funciones básicas:

a) Insertar pedido de producción (20 p).

Los parámetros de la función son el tipo de arduino a producir, la cantidad de unidades que se deben producir y el vector de producción.

Si el tipo de arduino existe en el vector de producción, incrementamos las unidades a producir.

Si el tipo de arduino no se encuentra en el vector, insertar el nuevo elemento.

b) Consultar producción de un Arduino (10 p)

Esta función permite ver la cantidad de unidades a producir para un tipo de arduino específico. La función posee como parámetros el tipo de arduino a consultar y el vector.

Arduino: BT
Cantidad a producir: 3000

c) Eliminar pedido de producción (20 p).

Esta función recibe como parámetros el **tipo** de arduino, la **cantidad** de unidades que se van a eliminar y el **vector**. Estas unidades se deben restar a las unidades que están asociadas al tipo de arduino. Si estas unidades a producir quedan en cero se debe **eliminar** el elemento del vector. Hay que **validar** que las unidades a eliminar no sobrepasen la cantidad a producir, de lo contrario dar un mensaje de error y volver al menú.

En el programa principal implemente un menú sobre la consola para que el usuario pueda usar esas funciones. Agregue estas opciones:

- consultar la lista de los tipos de arduinos que se pueden producir (10p),
- consultar la producción total (producción por cada arduino) (20p)
- opción para terminar el programa.

Este menú podría (20p) ser así (puede cambiarlo pero debe desplegar fecha y hora que tenga la computadora según se muestra):

```
=====
FÁBRICA DE ARDUINOS      28/3/2018 - 10:25:45 AM
```

```

      PRODUCCIÓN DE ARDUINOS
      CONTROL DE PEDIDOS
```

- 1- Insertar pedido de producción
- 2- Consultar producción de un arduino
- 3- Eliminar pedido de producción
- 4- Consultar tipos de arduinos que puede producir la fábrica
- 5- Consultar producción total
- 0- Fin

Opción seleccionada _

Puede agregar cualquier otra funcionalidad que vaya a mejorar el producto (el programa).

En el programa principal se deben solicitar al usuario los datos necesarios para luego ejecutar la función seleccionada:

- Para insertar y eliminar pedidos hay que solicitar al usuario el tipo de arduino y las unidades.
- Para consultar producción de un arduino hay que solicitar el tipo de arduino.
- Hay que validar que el tipo de arduino exista en el vector tipos_de_arduino, si el tipo dado por el usuario no está en esa lista hay que dar un mensaje de error y volver a pedir el tipo.

Inicialmente el vector de producción está vacío, luego se va actualizando conforme las funciones que realice el usuario.

La impresión o despliegue de la opción “Consultar producción total”, debe ser similar a lo siguiente (los tamaños de las columna a imprimir deben ser fijos, los tipos alineados a la izquierda en la columna TIPO, las cantidades y los % de producción alineadas a la derecha en las columnas respectivas, investigue sobre formatos de impresión):

```

                                PRODUCCIÓN TOTAL DE ARDUINOS
                                =====

TIPO                                CANTIDAD                                % PRODUCCIÓN

Arduino BT                          3000                                75
Arduino Industrial 101              1000                                25
-----
TOTAL A PRODUCIR                    4000                                100
```

NOTA: Cuando usamos estructuras definidas con <vector> o <array> en los parámetros de una función, estos pasan por valor.

Cuando ocupamos que los parámetros se pasen por referencia, en la definición de las funciones usamos el operador de dirección, símbolo &, como prefijo del parámetro. Esto es también una forma de retornar varios valores en una función: enviamos varias variables que van a ser actualizadas.

Ejemplo:

```
// si las variables par2 y par3 cambian, también cambian los argumentos
void funcion1(int par1, int &par2, int &par3)
```