**UNIVERSIDAD BLAS PASCAL**

**PROGRAMACIÓN PROCEDIMENTAL**

**SEMESTRE 2**

**2019**

**DESARROLLE LOS SIGUIENTES EJERCICIOS USANDO DEV C++.**

1. Un *número complejo* tiene la forma *c = (a , b\*i)* donde *a* y *b* son números reales y el factor ***i***es la unidad imaginaria, que se define como la raíz cuadrada del número *(‑1)*. Utilizando un registro con dos campos, puede representarse un número complejo, guardando en uno de los campos el número *a* (parte real) y en el otro el número *b* (parte imaginaria). El factor *i* (unidad imaginaria) se asume implícitamente en dicha representación, como acompañando al número *b* (y por lo tanto, no debe almacenarse en ninguna parte nada que represente a *i*). Desarrollar un programa que implemente las siguientes funciones diversas para realizar las siguientes operaciones de números complejos, en base a la representación sugerida en el párrafo anterior:
   1. Leer por teclado un número complejo.
   2. Visualizar un número complejo.
   3. Inicializar en cero un número complejo: (a,b) = (0,0)
   4. Sumar dos complejos: (a , b) + (c , d) = ( (a+c) , (b+d) )
   5. Restar dos complejos: (a , b) ‑ (c , d) = ( (a‑c) , (b‑d) )
   6. Multiplicar dos complejos: (a , b) \* (c , d) = ( (a\*c ‑ b\*d) , (a\*d + c\*b) )
   7. Devolver la parte real de un complejo: re( (a,b) ) = a
   8. Devolver la parte imaginaria de un complejo: im( (a,b) ) = b
   9. Devolver el valor absoluto de un complejo: abscom( (a,b) ) = sqrt ( a\*a + b\*b )
   10. Dar el conjugado de un complejo: conj((a,b)) = (a,‑b)
2. Escribir un programa que acepte la hora del día en *formato militar* (ejemplo: 1430), y encuentre la *representación usual* en horas y minutos, indicando si es AM o PM (mismo ejemplo: 14:30 PM). El programa también debe realizar la operación inversa, es decir, aceptar el formato usual, y convertirlo a formato militar. Para representar la hora en formato usual, puede usarse un registro con dos campos: uno para guardar la hora, y el otro para los minutos. La hora en formato militar será simplemente una variable de tipo *int*. Se sugiere que el programa contenga un menú de opciones, que permita elegir entre convertir de militar a usual, o convertir de usual a militar. Ejemplo:
   1. 0200 hora militar, es 2:00 AM hora usual.
   2. 3:45 PM hora usual, es 1545 hora militar.
3. Las coordenadas de un *punto en un plano* pueden guardarse en dos campos de un registro y de esta forma representar a dicho punto. Hacer un programa con menú de opciones que permita ingresar dos puntos *a* y *b*, representados como registros, y que contenga opciones para:
   * 1. Cargar los dos puntos por teclado.
     2. Mostrar los dos puntos en forma de par ordenado.
     3. Calcular la distancia de cada uno al origen de coordenadas (se aplica el teorema de Pitágoras...)
     4. La longitud del segmento de recta que los une: L= sqrt( (b.x ‑ a.x)2 + (b.y ‑ a.y)2 )
     5. La pendiente de la recta que los une: P = (b.y ‑ a.y) / (b.x ‑ a.x)
4. La hora de inicio de un acontecimiento cualquiera, puede ser manejada mediante un registro con tres campos: *Hora*, *Minutos*, y *Segundos*. Se desea desarrollar un programa que permita:
   1. Ingresar en un registro, la hora de inicio y la hora de fin de un acontecimiento, que calcule su duración neta, y que almacene la misma en un tercer registro. Esta duración neta, deberá ser presentada por pantalla, pero expresada en las cuatro posibilidades:
      1. Duración en horas, minutos y segundos
      2. Duración en horas
      3. Duración en minutos
      4. Duración en segundos
   2. Ingresar una hora cualquiera, pero como número real, y que la convierta a formato registro, redondeando los segundos y visualizando la misma por pantalla. Por ejemplo: la hora 13.5676 en formato real (o “flotante”), es la hora 13:34:03 en formato sexagesimal, redondeados los segundos.
5. Se desea cargar un arreglo de n registros, con la información de los artículos en stock de un comercio. Cada registro contiene los siguientes campos: código del artículo, descripción del artículo, cantidad actual en stock de ese artículo, cantidad mínima requerida para ese artículo, y precio unitario. Se pide:
   1. Determinar cuántos artículos del vector, tienen una cantidad actual en stock inferior a la cantidad mínima requerida para ese artículo.
   2. Determinar si en en el vector existe un artículo con código igual a x (x se carga por teclado). Si existe, cambiar su precio unitario por el valor pr (que también se carga por teclado). Si no existe, informar por pantalla con un mensaje.
   3. Generar otro vector a partir del primero, de forma que el nuevo vector contenga los datos de todos los registros del original en los que el valor de stock (o sea, cantidad actual \* precio unitario), sea mayor a m (que se carga por teclado). El registro del nuevo vector, debe tener los siguientes campos: código del artículo, descripción, y valor de stock.
6. (O) Un médico necesita tener almacenados ciertos datos sobre sus pacientes. Por cada paciente almacena lo siguiente: nombre, dirección, teléfono, fecha de la última visita, si tiene obra social o no, y si tiene alergias o no. Los datos deberán ser almacenados en un vector de registros y se desea un programa que ofrezca un menú con las siguientes opciones:
   1. Cargar el arreglo
   2. Visualizar un registro dado en pantalla (Consultas)
   3. Dado el nombre de un paciente, dar la fecha de su última visita.
   4. Dar nombre y teléfono de los pacientes con alergias.
   5. Dar alfabéticamente todos los pacientes que tengan Obra Social.
7. (O) Se desea almacenar el inventario de un comercio de artículos deportivos, en un arreglo de registros. La descripción de cada artículo, contiene el código del artículo, su descripción, la cantidad en stock y el precio unitario. Desarrollar un programa controlado por menú, que brinde opciones para:
   1. Ingresar n registros en el arreglo.
   2. Buscar un artículo, dado su código, y mostrar sus datos por pantalla. Esta opción también debe dar a elegir si el registro encontrado y visualizado deberá mantenerse en el arreglo, o eliminarse del mismo. En este último caso, por supuesto, la eliminación deberá realizarse. Si no se encuentra el registro, informar por pantalla.
   3. Visualizar todo el inventario, ordenado alfabéticamente según descripción de cada artículo
   4. Visualizar todo el inventario, ordenado decrecientemente por cantidad en stock.
8. Se necesita almacenar en un arreglo de registros, con fines estadísticos, la descripción de algunas personas. Cada registro deberá contener el nombre de la persona, su sexo, su edad, y su altura. Desarrollar un programa que lea y almacene n registros en este arreglo, lo ordene luego por orden alfabético de nombres, y visualice todo el contenido ya ordenado.
9. Desarrollar un programa que cargue dos arreglos: uno con registros de estudiantes y el otro con registros de empleados. Cada registro de estudiante debe tener campos para el apellido, primer nombre y una nota de una materia. Cada registro de empleado debe tener campos para apellido, nombre y salario. Ambos arreglos deben ordenarse alfabéticamente por apellido y por primer nombre. No pueden aparecer dos registros con el mismo nombre y apellido en el mismo arreglo. Por medio de una función, debe darse un aumento del 10% sobre su salario, a cada empleado sea además un estudiante (es decir, que figure en los arreglos) y cuya nota sea mayor a 7 (siete).
10. Una compañía de transporte de carga tiene 20 camiones, los cuales hacen entrega en diversos destinos del país. Para cada envío, se registra el número del camión que hace el transporte, el destino (un código del 0 al 19), el importe pagado por el envío, y el tipo de carga (un código ‘a’, ‘b’, o ‘c’).
    1. Cargar un vector de n componentes en memoria, en el cual cada componente sea una estructura describiendo cada envío.
    2. Determinar y mostrar el importe total obtenido por la compañía entre todos los envíos.
    3. Determinar qué porcentaje de envíos de cada tipo de carga (‘a’, ‘b’, o ‘c’) se realizaron.
    4. Determinar cuántos envíos hizo cada camión.
    5. Determinar el importe total obtenido por envíos al destino número 5.