UNIVERSIDAD NACIONAL DE INGENIERIA

EXAMEN FINAL DE PROGRAMACION ORIENTADA A OBJETOS
MB545

Tiempo: 110 min.
Fecha 26 de febrero del 2021

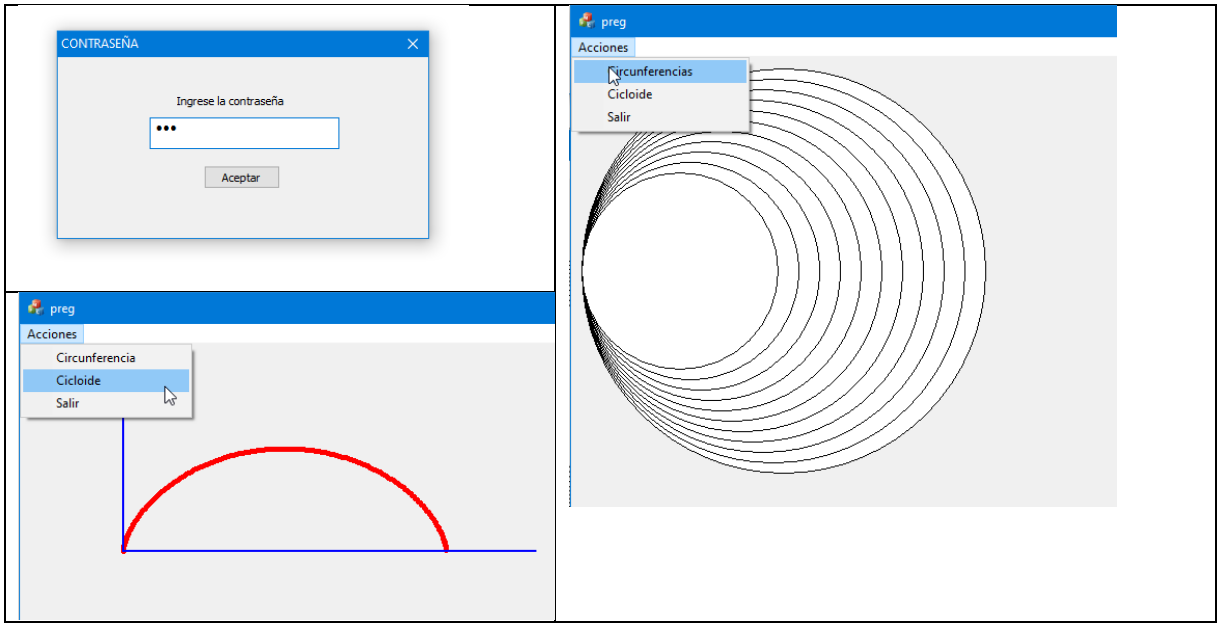
- Se monitoreará al alumno, y deberá compartir su pantalla cuando se le requiera
- Las soluciones no deben ser parecidas o iguales a uno o más alumnos se les **calificará con A0** a todos ellos.
- La solución de las preguntas 1 y 3 deberá pasar completas al word; las preguntas 2 y 4 solo copiar los códigos que agregó. Deberá agregar la corrida de cada problema como evidencia de que funciona. Luego subir todo como un único archivo pdf.
- La solución de la pregunta 2 y 4, además deberá comprimir la carpeta de archivos y subirlo. Antes debe eliminar todos los archivos de las carpetas que pesan más de 2 MB. La carpeta de la solución de cada pregunta debe ser menos de 2 MB.
- Es importante este punto por cuanto es la única forma de saber si el programa fue resuelto íntegramente y no fue copia.

Nota: Tendrá 30 minutos adicionales exclusivamente para que el alumno pueda subir las soluciones al aula virtual.

---

1. Se tiene la clase cRecta ubicado en el plano con coordenadas **aleatorias** A(a_1,a_2) y B(b_1,b_2), en el rango que Ud. considere, se pide la pendiente, distancia y la ecuación de dicha recta y resolverá su ecuación,use constructores.
   Derivará una clase llamada cPolígono, que permita hallar el área y el perímetro del cuadrilátero irregular, hallara las ecuaciones de las rectas que forman el cuadrilatero, las pendientes y sus longitudes.
   Tanto la clase cRecta como la clase derivada cPoligono con sus respectivos constructores. Si no tuviese los constructores, el puntaje asignado será de **3pts.** Con evidencia y repetición. Respuesta buena(5pts).

2. Desarrolle un programa en Visual C++, que inicie preguntando por una contraseña para ingresar al sistema, si el usuario indica la clave correcta, deberá volver a pedir la clave, hasta que ingrese la clave correcta. Una vez ingresado al sistema, el sistema debe incorporar un menú de 3 opciones, que permita Dibujar 11 circunferencias superpuestas de acuerdo a la muestra, dibujar una cicloide y Salir del sistema. En el Word y para el aula virtual, el desarrollo de esta pregunta deberá reportar únicamente los códigos agregados, procedimiento para agregar el cuadro de contraseña y el menú. Deberá asumir algunos datos no indicados. Y la evidencia al correo de su profesor, cuyas indicaciones se mencionan arriba (5pts)



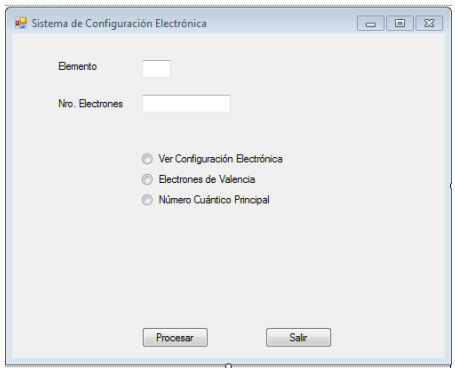
3. Una empresa tiene personal que labora para un determinado sector metal mecánica, desea un reporte de sus trabajadores y guardarlos en **un archivo binario, y visualizar en cualquier momento,** el programador piensa que puede usar una estructura para la información y asociarla con un **archivo binario**, para lo cual requiere los campos de: código, Apaterno, Amaterno, nombres, edad, dirección, Area de trabajo, HorasTrabajadas, tiempolabora, descuentos, bonificaciones y pago neto;
   En la empresa las horas normales de labor son de 40 horas semanales si superan estas horas hasta un tope de 10 horas se considera horas extras y se pagan 30% más de la tarifa normal que es de 20.00 soles la hora.
   El pago se realiza mensualmente y le descuentan por Afp 7.8% del SB, por salud 8.9% del SB. Por tiempo de servicio tienen una bonificación: de 5 hasta 10 años de 7.5% del SB, hasta 15 años de 8.5% del SB, y 9.8% de 20 años en adelante. El Gerente desea que, en la boleta, se registre los datos del empleado, el Área de trabajo, las Horas Laboradas, el Tiempo de Servicio, la Edad, el Total de Descuentos, la Bonificación y el Pago Neto. El ingreso de la información y el listado debe hacerse mediante un menú tal como se muestra, que da a criterio de cada alumno como confecciona el listado o la visualización de cada trabajador. Debe tener la evidencia de la compilación. (5pts).
   Boleta de los trabajadores
   <a>Nuevo Registro
   <b>Mostrar Registros
   <c>Modificar Horas
   <d>Salir

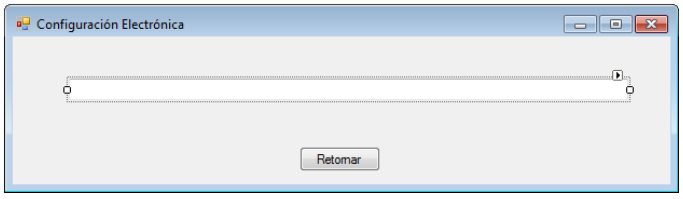
4. Aplicando Formularios en Visual C++ se le solicita ingresar el símbolo de un elemento químico de la Tabla Periódica con la cantidad de electrones respectiva (Usar una clase con atributos y métodos usando encapsulamiento). Y luego usando "radio button" pueda seleccionar para poder mostrar en otra(s) ventana(s) su configuración electrónica, sus electrones de valencia y/o el número cuántico principal (nivel

energético). La evidencia de la compilación se menciona arriba en las indicaciones generales. Si no cumple no tiene el puntaje asignado (5pts)
Usar como base la estructura de la siguiente clase:

```
class CElemento
        {
        protected:
        string elemento;
        int num_electrones;
        string configuracion;
        int num_valencia;
        int num_cuantico;
        public:
        void cargar(string _elemento, int _num_electrones);
        void genera_opc1();
        void genera_opc2();
        void genera_opc3();
};
```

Ejemplo: Si ingresa Elemento= "**Na**", Nro. Electrones= **11**. Y según la selección deber mostrar en otras ventanas (formularios):
Configuración Electrónica = "1s2, 2s2, 2p6, 3s1"
Electrones de valencia = 1
Número Cuántico Principal = 3





RUBRICA

CRITERIOS	5	3	2	1	0
CLARIDAD Y PRECISIÓN EN LA SOLUCION	La solución tiene evidencia del resultado con la corrida y lo guarda correctamente para su envío.	La solución no es completa, pero tiene corrida no hay evidencia.	No hay solución, la codificación no es completa, no tiene evidencia	No hay claridad en la codificación ni precisión	Respuesta con algunas líneas sin precisar
CONOCE Y RESUELVE LA PLATAFORMA VISUAL C++	Utiliza adecuadamente En modo consola la solución completa del problema planteado, usando las clases. E evidencia con la corrida	Resuelve en modo consola parcialmente la clase y la corrida no es completa, tiene evidencia	El resultado no demuestra el uso eficaz de la clase, no hay corrida	No hay solución del problema planteado, tiene líneas de código, pero no evidencia hacia una solución.	Sin solución ni líneas de código fuente
RESUELVE CON CONOCIMIENTO DE LA PLATAFORMA DE C++ CON WINDOWS	Usa una estrategia eficiente en la confección de formularios, ventanas y menú con opciones cumpliendo con lo requerido en la solución.	Usa una estrategia no adecuada acorde con la solución tiene errores subsanables, tiene corrida y evidencia sin llegar a la solución completa.	Usa una estrategia no entendible tiene errores de sintaxis. No concluye en la solución. Evidencia poca preparación	No Usa una estrategia eficiente, desconoce la solución tiene errores de sintaxis y no hay evidencia de solución	Sin solución ni código fuente evidencia no conocer el tema
CONOCE LA FORMA CORRECTA DE ENVIAR LA SOLUCION	Envía la solución en una carpeta a su drive del Gmail. Envía el link al profesor sin demora. Envía la solución en un archivo de Word o pdf al aula virtual de la solución del examen	Envía la solución al aula virtual. No envía el link a su profesor del curso.	Envía la solución solo parcialmente, no envía el total al aula ni al profesor	Envía pantallazos sin orden ni legibles	No envía nada desconoce

SOLUCIONARIO EXAMEN FINAL MB545 2020_II

```
//problema 1 para el final
#include <iostream>
#include <math.h>
#include <time.h>
#define pi 2*asin(1.0)
using namespace std;
class recta1{
     protected://construtores
     float a1, a2, b1, b2;
public:
     recta1();
     recta1(float,float,float,float);
     float pendiente();
     float distancia();
     void ecuacion();
     void imprimir();
};
recta1::recta1()
{a1=5;b1=10;a2=8;b2=15;}

recta1::recta1(float a11, float a22, float b11, float b22){

     a1 = a11;
     a2 = a22;
     b1 = b11;
     b2 = b22;
}
class poligono : public recta1 {
protected:
     float a3,b3,a4,b4;
public: // Metodos
     poligono();//constructores
     poligono(float,float,float,float,float,float,float,float
);
    float area();
    float perimetro();
};

// Constructor de la clase poligono
poligono::poligono()
{a1=10;a2=15;b1=20;b2=25;a3=18;b3=28;a4=30;b4=35;}
```

```
poligono::poligono(float m,float n,float p,float q,float a33,
float b33, float a44, float b44)
{
a1=m;
b2=n;
a2=p;
b2=q;
a3 = a33;
b3 = b33;
a4 = a44;
b4 = b44;
}



float recta1::pendiente() {
     float pendiente1;
    pendiente1=(b1-b2)*1.0/(a1-a2);
     return pendiente1;
}


float recta1::distancia() {
     float distancia1;
    distancia1=sqrt(pow(b1-b2,2)+pow(a1-a2,2));
     return distancia1;
}



void recta1::ecuacion() {
     float m, b;
     if((a1-a2)!=0){
    m=(b1-b2)*1.0/(a1-a2);
    b=a2-a1*m;
     cout <<" y = " <<m<<" x + "<<b<<endl;
     }
     else{
          cout<<"Error No es posible la operacion ";
          system("pause");
     exit(0);
     }
}


float poligono::area() {
    float area;
```

```
    area=(a1*b2+a2*b3+a3*b4+a4*b1)-(a2*b1+a3*b2+a4*b3+a1*b4);
    area=area/2;
    area=abs(area);
    return area;
}

float poligono::perimetro() {
    float
perimetro1,distancia_1,distancia_2,distancia_3,distancia_4;
    distancia_1=sqrt(pow(b1-b2,2)+pow(a1-a2,2));
    distancia_2=sqrt(pow(b2-b3,2)+pow(a2-a3,2));
    distancia_3=sqrt(pow(b3-b4,2)+pow(a3-a4,2));
    distancia_4=sqrt(pow(b1-b4,2)+pow(a1-a4,2));

perimetro1=distancia_1+distancia_2+distancia_2+distancia_4;
    return perimetro1;
}

void main() {
     srand((unsigned)time(NULL));

     float a1,a2,b1,b2,a3,b3,a4,b4;
     a1=rand()%23+1;
     a2=rand()%23+1;
     b1=rand()%23+1;
     b2=rand()%23+1;
     a3=rand()%23+1;
     b3=rand()%23+1;
     a4=rand()%23+1;
     b4=rand()%23+1;

     recta1 r1(a1,a2,b1, b2);
     recta1 r2(a2,a3, b2, b3);
     recta1 r3(a3,a4, b3, b4);
     recta1 r4(a1,a4,b1, b4);
     poligono pol(a1,a2,b1,b2,a3,b3,a4,b4);//
p(a1,a2,b1,b2,a3,b3,a4,b4);
     //PRUEBAS
     cout<<"La ecuacion de la recta 1 es: ";
     r1.ecuacion();
     cout<<"La ecuacion de la recta 2 es: ";
     r2.ecuacion();
     cout<<"La ecuacion de la recta 3 es: ";
     r3.ecuacion();
     cout<<"La ecuacion de la recta 4 es: ";
     r4.ecuacion();
```

```
     cout<<"La pendiente de la recta 1 es:
"<<r1.pendiente()<<endl;
     cout<<"La pendiente de la recta 2 es:
"<<r2.pendiente()<<endl;
     cout<<"La pendiente de la recta 3 es:
"<<r3.pendiente()<<endl;
     cout<<"La pendiente de la recta 4 es:
"<<r4.pendiente()<<endl;
     cout<<"La longitud de la recta 1 es:
"<<r1.distancia()<<endl;
     cout<<"La longitud de la recta 2 es:
"<<r2.distancia()<<endl;
     cout<<"La longitud de la recta 3 es:
"<<r3.distancia()<<endl;
     cout<<"La longitud de la recta 4 es:
"<<r4.distancia()<<endl<<endl;
     cout<<"El area del cuadrilatero es: "<<pol.area()<<endl;
     cout<<"El perimetro del cuadrilatero es:
"<<pol.perimetro()<<endl;

     system("pause");
}
*/

////////////////////////////////////
//problema 3 Para el Final 2020_II
#include<iostream>
#include<stdlib.h>
#include<fstream>
#include<time.h>
using namespace std;

struct  Boleta {
     long codigo;
     char Apaterno[40];
     char Amaterno[40];
     char Nombres[40];
     char Direccion[40];
     char Area[40];
     int HTrabajadas;
     float Tlabora;
     int Edad;
     double Descto,Bonif;
     double Sueldo;
};
typedef struct Boleta ficha;
float Ts;
```

```
char menu() {
	char op;
	system("cls");
	cout << "\n MENU " << endl
		<< "<a> Ingresar " << endl
		<< "<b> Mostrar Lista " << endl
		<< "<c> Eliminar " << endl
		<< "<d> Salir" << endl
		<< "OPCION--> "; fflush(stdin);
	cout << "Ingrese su opcion ";
	cin >> op;
	return op;
}

void insertar() {
	FILE* fich; Boleta regis;
	fich = fopen("Registros.bin ", "ab");
	if (fich == NULL) {
		cout << "El fichero no existe " << endl;
	}
	else {
		cout << "Ingrese Codigo ";
		cin>>regis.codigo;
		cout << "Ingrese su Apellido Paterno ";
fflush(stdin);
		cin.getline(regis.Apaterno, 40);
		cout << "Ingrese su Apellido Materno ";
fflush(stdin);
		cin.getline(regis.Amaterno, 40);
		cout << "Ingrese sus nombres "; fflush(stdin);
		cin.getline(regis.Nombres, 40);
		cout << "Ingrese su direccion "; fflush(stdin);
		cin.getline(regis.Direccion, 40);
		cout << "Ingrese Area de Trabajo "; fflush(stdin);
		cin.getline(regis.Area, 40);
		cout << "Ingrese sus horas trabajadas ";
		cin >> regis.HTrabajadas;
		cout<<"Ingrese Tiempo que labora ";
		cin>>regis.Tlabora;
		cout << "Ingrese su Edad ";
		cin >>regis.Edad;
		//cout<<"Ingrese el tiempo que labora ";
		//cin>>regis.Tlabora;
		if(regis.Tlabora>=5)
			Ts=40*20*0.075;
		if(regis.Tlabora<=10)
			Ts=40*20*0.085;
```

```
          if(regis.Tlabora>=20)
               Ts=40*20*0.098;


          int Hextras=0;
          if (regis.HTrabajadas <= 40)
               regis.Sueldo = 40*20-
(40*20*0.078+40*20*0.089)+Ts;  //(500) - (500*7.8/100.0) -
(500*8.9/100.0) ;

          else if (regis.HTrabajadas-40<=10)

          regis.Sueldo =((40*20)+40*20*0.30)-
(40*20*0.078+40*20*0.089)+Ts;
          //Descuentos
          regis.Descto=(40*20*0.078+40*20*0.089);
          //Bonific
          regis.Bonif=Ts;

          fwrite(&regis, sizeof(regis), 1, fich);// escribe
de acuerdo a la estrctura
          fclose(fich);//cierra el fichero y asi no quede
abierto
     }
}
void mostrar() {
     FILE* fich; ficha regis;
     fich = fopen("Registros.bin", "rb");

     if (fich == NULL) {
          cout << "El fichero no existe " << endl;
     }
     else {
          fread(&(regis), sizeof(regis), 1, fich);
          while (!feof(fich)) {
               cout << "\n_____________________";
               cout << "\nCodigo.............. " <<
regis.codigo    ;
               cout << "\nApellido Paterno..... " <<
regis.Apaterno;
               cout << "\nApellido Materno..... " <<
regis.Amaterno;
               cout << "\nNombres.............. " <<
regis.Nombres;
               cout << "\nDireccion............ " <<
regis.Direccion;
```

```
                cout << "\nArea.................. " <<
regis.Area;
                cout << "\nEdad.................. " <<
regis.Edad;
                cout << "\nHoras Trabajadas...... " <<
regis.HTrabajadas;
                cout << "\nBonificacion.......... " <<
regis.Bonif;
                cout << "\nDesccuento............ " <<
regis.Descto;
                cout << "\nSueldo a Pagar........ " <<
regis.Sueldo;


                cout << "\n_____________________" << endl;
                fread(&regis, sizeof(regis), 1, fich);
          }
     }
     fclose(fich);
     system("pause");
}

void main() {
     char op;
     do {
          op = menu();
          switch (op) {
          case 'a':
                insertar();
                break;
          case 'b':mostrar();
                break;
          }
     } while (op != 'd');
     system("pause");
}
```

Solución pregunta 2

```
BOOL CpregDlg::OnInitDialog()
{…….
       clave box;
       do{
       box.DoModal();
       } while (box.m_clave!=_T("123"));
…….

}
```

```
void CpregDlg::OnAccionesCircunferencia()
{ CClientDC g(this);
 for (int i=0;i<=10;i++)
  g.Ellipse(10,10+i*10,400-i*20,400-i*10);
}

void CpregDlg::OnAccionesCicloide()
{CClientDC g(this);
 CPen cpr;
 cpr.CreatePen(BS_SOLID,5,RGB(255,0,0));
 g.SelectObject(cpr);
 g.MoveTo(100,200);
 int x,y,r=50,np=200,pi=atan(1.0)*4;
 float t;
        for (int i=0;i<=np;i++)
        { t=i*2.0*pi/np;
          x=r*(t-sin(t));
          y=r*(1-cos(t));
          g.LineTo(100+x,200-y);
       }
 cpr.DeleteObject();cpr.CreatePen(BS_SOLID,2,RGB(0,0,255));
 g.SelectObject(cpr);
 g.MoveTo(100,200);
 g.LineTo(500,200);
 g.MoveTo(100,200);
 g.LineTo(100,50);
}

void CpregDlg::OnAccionesSalir()
{OnOK();
}
//////////////////////////////////////////
Solución P4
    #include <iostream>
    #include <string>
    using namespace std;
    class CElemento
    {
    private:
         string elemento;
         int    num_electrones;
         string configuracion;
         int    num_valencia;
         int    num_cuantico;

      public:
         void  Cargar(string _elemento, int _num_electrones);
         void  genera_opc1();
         void  imprimir();
    };



    #include "Elemento.h"
    void CElemento::Cargar(string _elemento, int
    _num_electrones){
            elemento = _elemento;
```

```
    num_electrones = _num_electrones;
}

void CElemento::genera_opc1(){

 // 1s2 2s2 2p6 3s2 3p6 4s2 3d10 4p6 5s2 4d10 5p6 6s2 4f14 5d10
6p6 7s2 5f14 6d10 7p6

  int M[3][19] = { {1, 2, 2, 3, 3, 4, 3, 4, 5, 4, 5, 6, 4, 5,
6, 7, 5, 6, 7},
                   {1, 1, 2, 1, 2, 1, 3, 2, 1, 3, 2, 1, 4, 3,
2, 1, 4, 3, 2},
                   {2, 2, 6, 2, 6, 2,10, 6, 2,10, 6, 2,14,10,
6, 2,14,10, 6} };

  num_cuantico = 0;
  num_valencia = 0;
  int cont_ele = 0;
  int j = -1;

  while (cont_ele < num_electrones)
  {
       j = j + 1;
       num_cuantico = M[0][j];
       if (cont_ele + M[2][j] < num_electrones) {
            cont_ele = cont_ele + M[2][j];
       } else {
            num_valencia = num_electrones - cont_ele;
            cont_ele = cont_ele + num_valencia;
       }


       string orbita = to_string(M[0][j]);
       switch (M[1][j])
       {
       case 1: orbita += "s"; break;
       case 2: orbita += "p"; break;
       case 3: orbita += "d"; break;
       case 4: orbita += "f"; break;
       }

       if (cont_ele < num_electrones) {
            orbita += to_string(M[2][j])+", ";
       }
       else
       {orbita += to_string(num_valencia);
       }

       configuracion += orbita;
```

```
  }

}

void CElemento::imprimir(){
  cout << "Elemento.................: " << elemento << endl;
  cout << "Numero electrones........: " << num_electrones <<
endl;
  cout << "Configuracion electronica: " << configuracion <<
endl;
  cout << "Numero de valencia.......: " << num_valencia <<
endl;
  cout << "Numero cuantico..........: " << num_cuantico <<
endl;
}
```