

**Enunciado del problema a resolver.** (Puntaje Total 7 (siete) sin errores. Tiempo máximo total 1h:30m).

**Se evalúan los temas de:** Archivos Binarios en C++ stream, Punteros y Estructuras Dinámicas de Datos +Temas Parcial 1.

Se requiere realizar un proceso que “**Informe las respuestas de los alumnos de cálculos de superficies de figuras geométricas**”; contando para ello con los siguientes archivos de datos:

1. **NombresFiguras.Dat:** sin orden, de 20 nombres de figuras geométricas (*str20*).
2. **RespuestasAlumnos.Dat:** ordenado por Nombre del Alumno con repetición, conteniendo cada registro:  
**2.a)** Nombre Alumno (*str10*), **2.b)** Es Respuesta Correcta (*bool*), **2.c)** Nombre Figura Geométrica (*str20*).

**Se deberá considerar el siguiente bloque principal:**

```
main() {
    // Definir tipos, variables, inicialización y argumentos a incluir en cada invocación a las funciones.
    Abrir(NombresFiguras,RespAlu);
    ProcNomFig(    ); // Asigna a Estr.Estatica nom. fig. geom. y NULL al puntero a Lista de Nombres de Alumnos y Ordena.
    ProcRespAlum(    ); // Debe emitir y contar resp.cor. y resp.incor. usando técnica de Corte de Control, etc.
    EmitirResulta2(    ); // Emitir un informe de cada nom.fig.geom. la cant. de resp.cor y resp.incor.
    Cerrar(NombresFiguras,RespAlu);
    return 0;
} // main
```

### Tabla de diseño de invocaciones y prototipos (interfaz) de funciones

<b>Constantes a utilizar:</b>
<b>const</b> CANT_FIGS = 20;
<b>Completar typedef, struct, arreglos y estructuras dinámicas de datos.</b>
<b>typedef</b> tLista; sFig tvrCalc [    ]; str20; str10; <b>struct</b> tInfo; sFig; sNodo; sResp;
<b>Prototipos de funciones cuyos módulos deberán desarrollarse con los argumentos adecuados:</b>
<b>void</b> ProcNomFig( <i>ifstream</i> &NomFG, tvrCalc vrCalc);
<b>void</b> ProcRespAlum( <i>ifstream</i> &RespAlu, tvrCalc vrCalc, <i>ofstream</i> &ExaAlu);
<b>void</b> EmitirResulta2( <i>tvrCalc</i> vrCalc, <i>ofstream</i> &ExaAlu);
<b>void</b> ReInsertaNodo( <i>tLista</i> &Lista, tLista &pNodo); // Ordenar por nomAlum.
<b>void</b> ReOrdenar( <i>tLista</i> &Lista);
<b>Prototipos de funciones que NO deberán desarrollarse pero si llamarlas con los argumentos adecuados:</b>
<b>void</b> InsertarEnOrden (tvrCalc vrCalc, sFig rFig, <b>short</b> card);
<b>short</b> BusBinVec( <i>tvrCalc</i> vrCalc, str20 nomFig, <b>short</b> ult);
<b>void</b> InsertaInicio( <i>tLista</i> &Lista, tInfo valor);
<b>void</b> SacarPrimerNodo( <i>tLista</i> &Lista, tInfo &valor);
<b>void</b> EnlazaInicio( <i>tLista</i> &Lista, tLista &pNodo);
<b>void</b> EnlazaEnMedio( <i>tLista</i> &Lista, tLista &pNodo);

**Se pide:**

1. (1 punto). Dibujar y codificar los tipos de datos: *tLista*, *sFig*, *sNodo*, *sResp*, *tInfo*, *tvrCalc* [ ], *str20*, *str10*. Además se deben indicar los argumentos y declaración de las variables y sus tipos del bloque principal. El dibujo debe ser: COMPLETO, PROLIJO, SIMPLE, CLARO, y con RÓTULOS APROPIADOS.

2. Desarrollar las siguientes funciones:

a) (2 puntos). Diagramar la función **ProcNomFig** que asigna los nombres de Fig.Geom. leídos de **NombresFiguras.Dat** y un puntero inicializado a NULL, invocando a la función **InsertarEnOrden** a una estructura de datos estática.

b) (2 puntos). Codificar la función **ProcRespAlum**, que lee los datos de **RespuestasAlumnos.Dat**. Esta función debe procesar los datos del archivo utilizando la técnica de Corte de Control, el cual deberá sumar 1 por cada registro leído, si respondió bien en una variable o si respondió mal en otra variable y además en el **nodo** de la lista asignar en un registro al campo nombre del alumno y si respondió bien (valor 1) o mal (valor 0), al campo resp a la fig.geom. correspondiente. Esta función debe invocar a **BusBinVec** para colgar el nodo según la fig.geom. y a **InsertInicio**. También deberá emitir títulos apropiados, el nombre del alumno evaluado, la cantidad de respuestas correctas e incorrectas y un mensaje apropiado si la cantidad de resp.cor. es mayor a la cantidad de resp.incor. "Excelente", sino, "Lo Lamento". El diseño del listado es:

**Exámenes de Superficies de Figuras Geometricas ordenado por nombre Alumno**

Alumno/a	cResp.Cor.	cResp.InCor.	Evaluacion
X(20)	9	9	Lo lamento
X(20)	9	9	Excelente

**Observación:** Si la cResp.InCor. es menor a la cResp.Cor. el mensaje de evaluación debe ser *Excelente*, sino, debe ser *Lo lamento*.

c) (2 puntos). Diagramar o Codificar la función **EmiteResulta2** de las cantidades totales de respuestas correctas y de respuestas incorrectas por cada una de las figuras geométricas, el cual debe **ReOrdenar** cada una de las listas por el nombre del alumno ascendente que a su vez invoca a **ReInsertaNodo**. Se debe invocar a la función **SacarPrimerNodo**. El diseño del informe es:

**Listado cant. Resp.Cor. e InCor. de Alumnos ordenado por Figuras Geométricas y nombre Alumno**

Nombre Figura X(20)		
Nombre del alumno/a	Respuesta	
X(20)	1 bien	Repita para otras figuras
X(20)	0 mal	
.	.	
Total Figura: x(20)	bien: 99 mal: 99	
Tot.Resp.Correctas : 99		Por fin de Proceso
Tot.Resp.Incorrectas: 99		

**OBSERVACIONES**

- (1) El enunciado se complementa con explicaciones adicionales y gráficos indicados en la pizarra.
- (2) **IMPORTANTE:** Resolver primero el punto 1; tiempo estimado 15 minutos.
- (3) Establecer una muestra de datos de los archivos de datos.
- (4) Respetar el orden de los campos de los archivos binarios según se indicó más arriba.
- (5) Las salidas de los resultados, debe estar dirigida al archivo de texto.
- (6) Dado que el **uso de los ciclos afecta el rendimiento del proceso**, se considerará oportuno optimizar estas situaciones de las estructuras de control de programas de la **repetición**, como así también de la **selección** y la **concatenación**.

**S.E.ú O.**

**Afirmaciones** (Tiempo máximo total 5 min. puntaje máximo 3 (tres) puntos)

**Marcar con un círculo** la afirmación que ud. considera correcta **V** o **F**, si responde bien, suma 1, y si NO responde, o responde mal, suma 0. **NO escribir en la columna Pntj.**

Realizar una prueba de escritorio o no (usando lápiz y papel), para los ejercicios con código.

	Afirmación	Marcar		Pntj.
1	<p><u>Dado el siguiente código:</u> La sobrecarga funcional permite utilizar el mismo nombre para distintas funciones, con argumentos que difieran en el orden, cantidad o tipos de datos a cada función, aunque, para cada una de ellas se debe repetir el código.</p>	V	F	
2	<p><u>Dado el siguiente código:</u> (Puede realizar la <u>Prueba de Escritorio</u>)  <b>template &lt;typename T&gt;</b>  <b>struct</b> sNodo {              T info;              sNodo&lt;T&gt;* sgte;          };  <b>template &lt;typename T&gt;</b>  <b>void</b> Push(sNodo&lt;T&gt;* &amp;Pila, T valor) {              sNodo&lt;T&gt;* pNodo;                pNodo = <b>new</b> sNodo&lt;T&gt;;              pNodo-&gt;Info = valor;              pNodo-&gt;Sgte = Pila;              Pila = pNodo;          } // Push          Permite invocar a la función Push con distintos tipos de datos, utilizando el mismo código.          Push(Pila1, 34); Push&lt;char&gt;(Pila2, 'a'); Push(Pila3, "hola"); Push(Pila4, rArt);  <u>siendo:</u>          sNodo&lt;short&gt;* Pila1, sNodo&lt;char&gt;* Pila2, sNodo&lt;str20&gt;* Pila3, sNodo&lt;sArt&gt;* Pila4.</p>	V	F	
3	<p><u>Dado el siguiente código:</u> (Debe realizar la <u>Prueba de Escritorio</u>)  <b>int</b> mcd(<b>int</b> x, <b>int</b> y) {              <b>if</b> (y == 0)                  <b>return</b> x;              <b>else</b> {                  <b>return</b> mcd(y, x % y);              }          }          } //mcd cout &lt;&lt; "mcd(36, 8) = " &lt;&lt; mcd(36,8); // Emite 4.</p>	V	F	

**Prueba de Escritorio para la afirmación 3.**

x      y      Stack del sistema      retorna

**Afirmaciones** (Tiempo máximo total 5 min. puntaje máximo 3 (tres) puntos)

**Marcar con un círculo** la afirmación que ud. considera correcta **V** o **F**, si responde bien, suma 1, y si NO responde, o responde mal, suma 0. **NO escribir en la columna Pntj.**  
 Realizar una prueba de escritorio o no (usando lápiz y papel), para los ejercicios con código.

	Afirmación	Marcar	Pntj.
1	<p><u>Dado el siguiente código:</u>                      La sobrecarga funcional permite utilizar el mismo nombre para distintas funciones, con argumentos que difieran en el orden, cantidad o tipos de datos a cada función, aunque, para cada una de ellas se debe repetir el código.</p> <p><b>VERDADERO</b>                      El código de cada una de estas funciones de igual nombre puede ser exactamente el mismo o bien puede diferir. En cambio con el uso de template, simplifica el código escribiéndolo solo una vez y se adapta a distintos tipos de datos, aunque el código es siempre el mismo.</p>	V F	
2	<p><u>Dado el siguiente código:</u> (Puede realizar la Prueba de Escritorio)  <b>template &lt;typename T&gt;</b>  <b>struct</b> sNodo {                          T info;                          sNodo&lt;T&gt;* sgte;                  };  <b>template &lt;typename T&gt;</b>  <b>void</b> Push(sNodo&lt;T&gt;* &amp;Pila, T valor) {                          sNodo&lt;T&gt;* pNodo;                          pNodo = <b>new</b> sNodo&lt;T&gt;;                          pNodo-&gt;Info = valor;                          pNodo-&gt;Sgte = Pila;                          Pila = pNodo;                  } // Push                      Permite invocar a la función Push con distintos tipos de datos, utilizando el mismo código.                      Push(Pila1, 34); Push&lt;char&gt;(Pila2, 'a'); Push(Pila3, "hola"); Push(Pila4, rArt);  <u>siendo:</u>                      sNodo&lt;short&gt;* Pila1, sNodo&lt;char&gt;* Pila2, sNodo&lt;str20&gt;* Pila3, sNodo&lt;sArt&gt;* Pila4.</p> <p><b>VERDADERO</b></p>	V F	
3	<p><u>Dado el siguiente código:</u> (Debe realizar la Prueba de Escritorio)  <b>int</b> mcd(<b>int</b> x, <b>int</b> y) {                          <b>if</b> (y == 0)                              <b>return</b> x;                          <b>else</b> {                              <b>return</b> mcd(y, x % y);                          }                  } //mcd cout &lt;&lt; "mcd(36, 8) = " &lt;&lt; mcd(36,8); // Emite 4.</p> <p><b>VERDADERO</b>                      36 / 8 resto es : 4; luego 8 / 4 resto es: 0, finalmente x = 4 cuando resto es cero. La función retorna 4 que es el mcd.</p>	V F	

**Prueba de Escritorio para la afirmación 3.**

x y Stack del sistema retorna  
**Folleto explicativo del problema a resolver**

## ESTRUCTURAS DE DATOS

#define record struct

**Archivo: NombresFiguras.Dat**

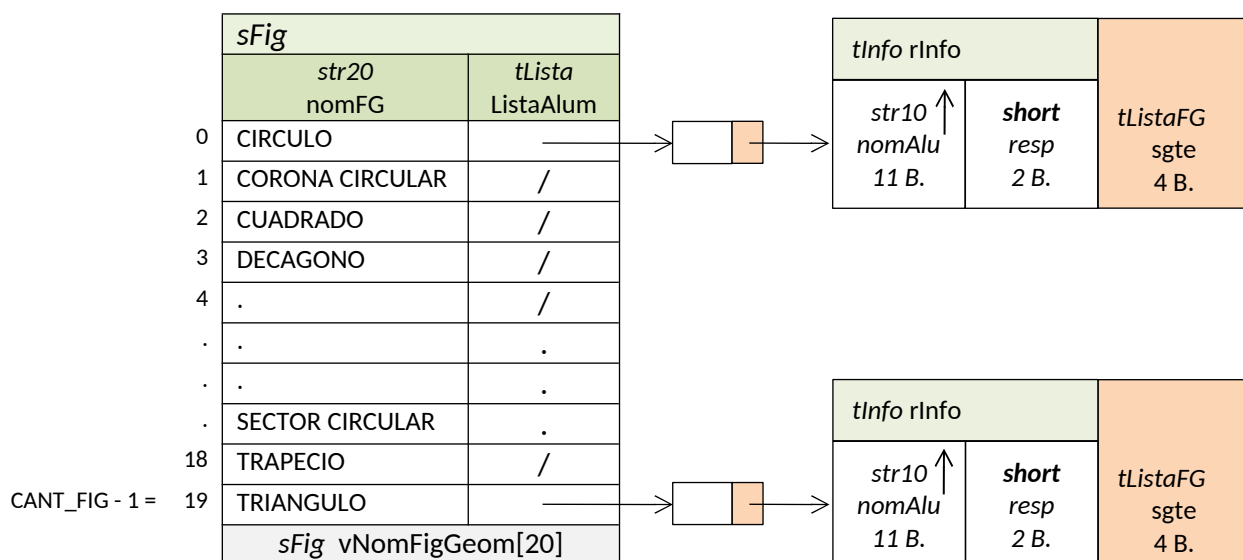
Contiene los nombres de 20 figuras geométricas desordenados.

**Archivo: RespuestasAlumnos.Dat**

**Representación gráfica del registro del archivo**

<i>sResp</i>		
<i>str10</i> nomAlu 11b.	<i>bool</i> EsRespCor 1b.	<i>str20</i> nomFig 21b.

**Representación gráfica del vector de punteros a listas de cada Figura Geométrica.**



El vector con datos se asignan, al recorrer secuencialmente el archivo de **NombresFiguras**, asignando al campo *nomFig* el dato leído y en el campo *ListaAlu* el valor NULL. Luego se ordena por el campo *nomFG*. Al recorrer secuencialmente el archivo de **RespuestasAlumnos** por cada registro leído se genera un nodo colgado de la componente según la figura geométrica leída, asignando al nodo el *nomAlu*, y en el campo *resp* el valor 0 si respondió mal o el valor 1 si respondió bien, dato leído del archivo indicado del campo *EsRespCor*.

Luego del Listado **ProcRespAlum** y antes de hacer el Listado **EmiteResulta2**, se deben ReOrdenar cada una de las Listas por el campo *nomAlu*.

**Exámenes de Superficies de Figuras Geometricas**

Alumno/a	cResp.	Cor. cResp.	InCor. Evaluacion
IGNACIO	3	1	Excelente
LUIS	2	4	Lo lamento
MARCOS	4	1	Excelente
MARISABEL	6	2	Excelente
PABLO	5	1	Excelente
PATRICIA	3	4	Lo lamento
PEDRO	7	0	Excelente
RICARDO	3	3	Lo lamento
ROCIO	6	0	Excelente
TERESA	5	0	Excelente

**Listado cant.Resp.Cor.Incor.x Figuras Geometricas****CIRCULO**

Nombre Alumno Respuesta

MARISABEL 1 bien

PATRICIA 0 mal

PEDRO 1 bien

TERESA 1 bien

Total Figura: CIRCULO bien: 3 mal.: 1

**CORONA CIRCULAR**

Nombre Alumno Respuesta

IGNACIO 1 bien

MARCOS 0 mal

PABLO 1 bien

Total Figura: CORONA CIRCULAR bien: 2 mal.: 1

**CUADRADO**

Nombre Alumno Respuesta

MARCOS 1 bien

PATRICIA 0 mal

Total Figura: CUADRADO bien: 1 mal.: 1

**DECAGONO**

Nombre Alumno Respuesta

LUIS 1 bien

PEDRO 1 bien

ROCIO 1 bien

Total Figura: DECAGONO bien: 3 mal.: 0

**DELTOIDE**

Nombre Alumno Respuesta

ROCIO 1 bien

TERESA 1 bien

Total Figura: DELTOIDE bien: 2 mal.: 0

**DODECAGONO**

Nombre Alumno Respuesta

LUIS 1 bien

MARISABEL 0 mal

RICARDO 0 mal

ROCIO 1 bien

Total Figura: DODECAGONO bien: 2 mal.: 2

**ELIPSE**

Nombre Alumno Respuesta

LUIS 0 mal

MARCOS 1 bien

PABLO 1 bien

PATRICIA 1 bien

PEDRO 1 bien

Total Figura: ELIPSE bien: 4 mal.: 1

**ESTRELLA 4 PUNTAS**

Nombre Alumno Respuesta

RICARDO 0 mal

Total Figura: ESTRELLA4PUNTAS bien: 0 mal.: 1

**HEPTAGONO**

Nombre Alumno Respuesta

MARISABEL 1 bien

PABLO 1 bien

PEDRO 1 bien

Total Figura: HEPTAGONO bien: 3 mal.: 0

**HEXAGONO**

Nombre Alumno Respuesta

IGNACIO 1 bien

LUIS 0 mal

MARISABEL 1 bien

PEDRO 1 bien

RICARDO 0 mal

TERESA 1 bien

Total Figura: HEXAGONO bien: 4 mal.: 2

**OCTOGONO**

Nombre Alumno Respuesta

MARISABEL 1 bien

Total Figura: OCTOGONO bien: 1 mal.: 0

**OVALO**

Nombre Alumno Respuesta

MARISABEL 1 bien

Total Figura: OVALO bien: 1 mal.: 0

**PARALELOGRAMO**

Nombre Alumno Respuesta

LUIS 0 mal

ROCIO 1 bien

TERESA 1 bien

Total Figura: PARALELOGRAMO bien: 2 mal.: 1

**PENTAGONO**

Nombre Alumno Respuesta

PATRICIA 1 bien

RICARDO 1 bien

Total Figura: PENTAGONO bien: 2 mal.: 0

**RECTANGULO**

Nombre Alumno Respuesta

PATRICIA 1 bien

Total Figura: RECTANGULO bien: 1 mal.: 0

**ROMBO**

Nombre Alumno Respuesta

MARCOS 1 bien

MARISABEL 1 bien

PATRICIA 0 mal

Total Figura: ROMBO bien: 2 mal.: 1

**ROMBOIDE**

Nombre Alumno Respuesta

IGNACIO 0 mal

MARISABEL 0 mal

PABLO 0 mal

PEDRO 1 bien

RICARDO 1 bien

Total Figura: ROMBOIDE bien: 2 mal.: 3

**SECTOR CIRCULAR**

Nombre Alumno Respuesta

MARCOS 1 bien

Total Figura: SECTOR CIRCULAR bien: 1 mal.: 0

**TRAPEZOIDE**

Nombre Alumno Respuesta

IGNACIO 1 bien

LUIS 0 mal

PABLO 1 bien

RICARDO 1 bien

ROCIO 1 bien  
Total Figura: TRAPEZOIDE bien: 4 mal.: 1

## TRIANGULO

Nombre Alumno Respuesta

PABLO 1 bien

PATRICIA 0 mal

PEDRO 1 bien

ROCIO 1 bien

TERESA 1 bien

Total Figura: TRIANGULO bien: 4 mal.: 1

**Tot.Resp.Correctas : 44**

**Tot.Resp.InCorrectas: 16**