

**FUNDAMENTOS DE PROGRAMACIÓN**  
**EXAMEN PARCIAL**  
**SEMESTRE ACADÉMICO 2021-2**

Horarios: Todos

Duración: 180 minutos

Revisado por los profesores del curso.

**INDICACIONES DEL EXAMEN**

- El examen parcial se realizará el día viernes 22 de octubre desde las 15:00 horas hasta las 18:00 horas.
- La rúbrica para la corrección del examen se encontrará publicada desde la semana anterior al examen en la plataforma PAIDEIA, en el curso 2021-2 FUNDAMENTOS DE PROGRAMACIÓN (1INF01), sección "Examen Parcial".
- Los enunciados de las preguntas se irán habilitando, en forma progresiva, en la plataforma PAIDEIA en el curso 2021-2 FUNDAMENTOS DE PROGRAMACIÓN (1INF01) en la sección "Examen Parcial"
  - El día viernes 22 de octubre, se ocultarán todas las secciones del curso en la plataforma PAIDEIA desde las 12:00 horas hasta las 19:00 horas, quedando sólo habilitada la sección del examen respectivo. No se podrá tener acceso a material alguno durante el rango de horas indicado el día del examen. Esto con el fin de que la sección "Examen Parcial" sea ubicada con facilidad.
  - Ninguna información relacionada al examen parcial aparecerá en las secciones de PAIDEIA de los cursos por horario. Estas secciones no se usarán durante el desarrollo del examen.
- El examen debe rendirlo en un computador que tenga el software PSeInt para desarrollar pseudocódigos, así como un Dev-C++ o Visual Studio Code, para desarrollar programas en lenguaje C. Solo se puede usar Visual Studio Code si cuenta con sistema operativo macOS. Es su responsabilidad verificar que su computador y los softwares requeridos funcionen correctamente. Se recomienda realizar las pruebas necesarias para verificarlo.
- Debe verificar, el día anterior al examen, que su computador tenga instalada la última versión de Zoom.
- Debe verificar que el ingreso a Zoom se realice con la cuenta Zoom asociada al usuario PUCP. Si se conecta como invitado a Zoom no ingresará a la sesión del examen.
- En la sección "Examen Parcial" en la plataforma PAIDEIA en el curso 2021-2 FUNDAMENTOS DE PROGRAMACIÓN (1INF01), se encontrarán los enlaces para poder iniciar la sesión en Zoom. Esta se encontrará habilitada desde 10 minutos antes de la hora de inicio del examen. Las sesiones de Zoom están separadas por intervalo de códigos de alumnos, NO POR HORARIO. Debe ingresar a la sesión de Zoom del rango de códigos donde se encuentre su código de alumno.
- El día del examen a la hora programada, dentro de la sección "Examen Parcial", se encontrarán 3 tareas configuradas para cada pregunta (Pregunta 1, Pregunta 2 y Pregunta 3). En cada una de estas tareas, se encontrará el enunciado correspondiente. Deberá asegurarse que su computador pueda leer archivos en formato PDF. Este formato es el mismo que se ha usado para las presentaciones del curso, los enunciados de problemas y los enunciados de laboratorio durante todo el semestre.
- Cada alumno deberá conectarse a la plataforma Zoom 10 minutos antes de la hora programada del examen. Al ingresar debe colocar su código y nombre en el chat. El nombre del usuario de Zoom debe ser su nombre y apellidos.
- Cuando el Jefe de Laboratorio le indique, deberá iniciar la grabación con el Zoom, aceptar la invitación que se le envíe en forma inmediata para acceder a su sala personal. En la sala personal, debe compartir todo su escritorio, tener el micrófono y la cámara activos. La cámara debe enfocar en todo momento el rostro completo de frente del alumno, el cual debe mostrarse en forma nítida. El alumno no puede usar audífonos, a excepción que se le permita utilizarlo en los laboratorios.
- El alumno debe tener el mismo comportamiento que tendría en una evaluación presencial.
- El Jefe de Laboratorio le solicitará verificar su identidad, para lo cual el alumno debe mostrar a la cámara su DNI, TI o otro documento de identidad con fotografía.
- Los jefes de laboratorio, estarán cuidando el examen y estarán visitando las salas. A los alumnos que no compartan todo su escritorio, no tengan su micrófono y cámara activos NO SE LES CORREGIRÁ EL EXAMEN, y se les pondrá la nota cero (0). Ante cualquier problema técnico que pueda existir para compartir el escritorio, deberá comunicarse con el jefe de laboratorio.
- Si el alumno necesita usar una hoja durante el desarrollo del enunciado, puede únicamente usar una hoja en blanco; pero antes de ello deben llamar al Jefe de Laboratorio para mostrar dicha hoja a la cámara y él autorizará el uso de esta. A los alumnos que usen una hoja en blanco se les supervisará más seguido durante el examen.
- No está permitido utilizar otros dispositivos ajenos al computador donde se resuelve la evaluación, por ejemplo, tablet, celular, otra laptop u otro computador. Este incumplimiento se analizará como un caso de plagio.
- Si se detectan posibles casos de plagio, cada jefe de laboratorio notificará a los profesores y la junta de profesores revisará cada caso. En caso que la junta de profesores compruebe la falta, se procederá a anular el examen, a comunicar a EEGCC y se proseguirá como indica el "Reglamento Unificado de Procedimientos Disciplinarios de la Universidad".

## INDICACIONES DEL EXAMEN (2):

- **Durante el desarrollo del examen, mientras desarrolla cada pregunta debe explicar oralmente con sus palabras lo que está realizando. En caso de no hacerlo, se le asignará la nota cero (0) en la pregunta respectiva.**
- Cada pregunta del examen parcial, deberá resolverse en 50 minutos. Pasados los 50 minutos por pregunta, el alumno tendrá 10 minutos para subir sus archivos a la plataforma PAIDEIA:
  - La pregunta 1 aparecerá en la plataforma PAIDEIA desde la hora del inicio del examen, es decir desde las 15:00 horas. Existirán diversas variantes de enunciados para la pregunta, la identificarán fácilmente por que se le agregará una letra al número de la pregunta, por ejemplo “Pregunta 1A”, “Pregunta 1B”, y así sucesivamente, dependiendo de la pregunta que les toque (a cada alumno le tocará una pregunta 1 de manera aleatoria). El alumno tendrá hasta las 15:50 hrs para responder esta pregunta. Los siguientes 10 minutos, el alumno deberá dedicarlos a subir el archivo con su solución a la plataforma PAIDEIA (desde las 15:50 hrs. hasta las 15:59 hrs). El archivo debe ser grabado con el nombre **E1\_código\_pregunta1.psc** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta1.psc**, sin espacios). La plataforma PAIDEIA NO ACEPTARÁ archivos luego de dicho lapso de tiempo (a las 16:00 hrs no se permitirá la subida de archivos de esta pregunta). La solución de la pregunta 1 debe ser subida en la tarea correspondiente a la pregunta 1, sólo así la pregunta será considerada en la calificación. **Antes de subir la pregunta 1 a PAIDEIA debe asegurarse que se ha grabado correctamente en PseInt, es responsabilidad del alumno verificar que el archivo contenga lo que desarrolló. No se aceptará ningún pedido de corrección debido a que la pregunta no tiene contenido.**
  - La pregunta 2 aparecerá en la plataforma PAIDEIA desde el inicio de la segunda hora del examen, desde las 16:00 hrs. Existirán diversas variantes de enunciados para la pregunta, la identificarán fácilmente por que se le agregará una letra al número de la pregunta, por ejemplo “Pregunta 2A”, “Pregunta 2B”, y así sucesivamente, dependiendo de la pregunta que les toque (a cada alumno le tocará una pregunta 2 de manera aleatoria). El alumno tendrá hasta las 16:50 hrs para responder esta pregunta. Los siguientes 10 minutos, el alumno deberá dedicarlos a subir el archivo con su solución a la plataforma PAIDEIA (desde las 16:50 hrs. hasta las 16:59 hrs). El archivo debe ser grabado con el nombre **E1\_código\_pregunta2.c** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta2.c**, sin espacios). La plataforma PAIDEIA NO ACEPTARÁ archivos luego de dicho lapso de tiempo (a las 17:00 hrs no se permitirá la subida de archivos de esta pregunta). La solución de la pregunta 2 debe ser subida en la tarea correspondiente a la pregunta 2, sólo así la pregunta será considerada en la calificación.
  - La pregunta 3 aparecerá en la plataforma PAIDEIA desde el inicio de la tercera hora del examen, desde las 17:00 hrs. Existirán diversas variantes de enunciados para la pregunta, la identificarán fácilmente por que se le agregará una letra al número de la pregunta, por ejemplo “Pregunta 3A”, “Pregunta 3B”, y así sucesivamente, dependiendo de la pregunta que les toque (a cada alumno le tocará una pregunta 3 de manera aleatoria). El alumno tendrá hasta las 17:50 hrs para responder esta pregunta. Los siguientes 10 minutos, el alumno deberá dedicarlos a subir el archivo con su solución a la plataforma PAIDEIA (desde las 17:50 hrs. hasta las 17:59 hrs). El archivo debe ser grabado con el nombre **E1\_código\_pregunta3.c** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta3.c**, sin espacios). La plataforma PAIDEIA NO ACEPTARÁ archivos luego de dicho lapso de tiempo (a las 18:00 hrs no se permitirá la subida de archivos de esta pregunta). La solución de la pregunta 3 debe ser subida en la tarea correspondiente a la pregunta 3, sólo así la pregunta será considerada en la calificación.
- Todas las preguntas que se han propuesto han sido revisadas por todos los profesores del curso. Se está garantizando que: todas las preguntas puedan ser solucionadas en menos de 50 minutos y que todas las preguntas abarquen los temas vistos en clase.
- Durante la evaluación solo se puede tener abierto el Zoom, plataforma PAIDEIA, un visor de PDF, PseInt o el IDE que se utilice para desarrollar los pseudocódigos o programas y Discord. En Discord solo puede tener abiertos los canales definidos para realizar consultas sobre las preguntas del examen. No está permitido el uso de ningún material, archivo u otro aplicativo (correo, WhatsApp, Facebook, Twitter, Calculadora, Spotify u otro diferente), tampoco puede abrir otra pestaña en el navegador. Debe apagar su teléfono móvil y tenerlo cerca para mostrarlo al Jefe de Laboratorio si se lo solicita. El incumplir estas indicaciones, será motivo de anulación del examen.
- Está terminantemente prohibido tener comunicación con cualquier persona que no sea el Jefe de Laboratorio o el profesor, sin importar el medio utilizado. De igual manera, se prohíbe también la comunicación entre alumnos, sin importar el medio utilizado. Si se demuestra que dos o más alumnos se han comunicado durante el examen, será motivo para la anulación total de los exámenes pertenecientes a los alumnos involucrados.
- No está permitido compartir total o parcialmente soluciones del examen. Si se identifican dos o más desarrollos iguales, se procederá a anular la totalidad del examen de todos los alumnos involucrados. La Coordinación del Curso aplicará las herramientas disponibles para encontrar desarrollos iguales entre las soluciones presentadas por los alumnos.
- El profesor del curso puede en cualquier momento ingresar a la sala personal del alumno y hacerle alguna consulta sobre el desarrollo que usted está realizando. El objetivo de esta consulta es verificar que el alumno es quién está resolviendo el problema, por lo tanto, debe estar en la capacidad de explicar la implementación que realiza. En caso sus respuestas no evidencien que el alumno está desarrollando el examen, el profesor lo reportará a la Coordinación del Curso quién analizará el caso y determinará si se corregirá el examen o se anulará la totalidad de este.
- **Se considera presente y se califica el examen a los alumnos que se conectan a Zoom, que graban el desarrollo de todo el examen y suben la grabación al Classroom. También se considera presente a los alumnos que: descargan cualquiera de los enunciados de las preguntas de PAIDEIA aunque no se conecten a Zoom, que no realizan la grabación de todo el examen o no suben la grabación a Classroom; a estos alumnos se les asignará la nota cero (0), porque no es posible comprobar que fueron ellos quiénes desarrollaron el examen.**
- En caso de que usted tuviera algún problema de conexión. Deberá comunicarse inmediatamente con la Coordinación del Curso, con la profesora Silvia Vargas a través del correo [silvia.vargas@pucp.edu.pe](mailto:silvia.vargas@pucp.edu.pe). Deberá guardar evidencia del problema que está teniendo. Se consideran evidencias: fotos, videos, pantallas de error, etc. En las evidencias debe quedar claramente definida la hora y la fecha de lo ocurrido. Para analizar el caso es necesario que el alumno presente las evidencias, si no se envían evidencias el caso no se analizará. El correo debe tener el siguiente formato:

### INDICACIONES DEL EXAMEN (3):

- Asunto: 11NF01 - Problema de conexión para ingresar a Zoom o a la plataforma PAIDEIA para el Examen Parcial.
- Cuerpo del correo:
  - Horario:
  - Código:
  - Apellidos:
  - Nombres:
  - Descripción del problema:
- Adjuntar evidencias: fotos, videos o pantallas de error; donde se visualice la fecha y hora del problema de conexión.
- **En caso el correo no siga el formato establecido, no se analizará la solicitud.**
- Los correos recibidos se responderán durante la semana siguiente a la semana del examen.
- No se considera problema de conexión no subir el archivo dentro del tiempo establecido, por haberlo usado para continuar con la solución del problema. Tampoco se considera error de conexión, no saber usar la plataforma PAIDEIA o las herramientas PSEINT, DEV++ o VISUAL STUDIO CODE.
- **La Coordinación del Curso enviará un correo a los alumnos con el enlace correspondiente a Classroom para el examen, en el cual debe subir la grabación del examen correspondiente. El plazo para subir la grabación es hasta el día domingo 24 de octubre a las 23:59 horas. Si no sube la grabación, no se podrá corregir el examen, por lo cual se asignará la nota cero (0). De tener inconvenientes con la grabación debe escribir un correo a la Coordinación del Curso indicando los problemas que existen con la grabación respectiva y adjuntando las evidencias que lo sustenten. En caso no adjunte la evidencia, no se analizará el caso y se asignará la nota cero (0). No se responderá ningún mensaje enviado por Classroom.**
- Durante el proceso de corrección, el profesor puede determinar que requiere consultar al alumno sobre la solución que realizó. En este caso, el profesor escribirá un correo al alumno para agendar una reunión por Zoom, a esta reunión el alumno debe asistir puntualmente, compartir todo su escritorio, tener el micrófono y la cámara activos, y responder las preguntas que realice el profesor. En caso las respuestas no evidencien que el alumno es el que desarrolló el examen, el profesor lo comunicará a la Coordinación del Curso quién analizará el caso y determinará si se corregirá la pregunta o se anulará todo el examen.

### SOBRE DUDAS Y ACLARACIONES RESPECTO A LOS ENUNCIADOS:

- Si, durante la evaluación, tiene alguna duda sobre el enunciado, debe realizarla a través del canal de Discord específico para la pregunta, al cual debe estar conectado desde su computador. El servidor de Discord será creado por la Coordinación del Curso previo al examen y se le comunicará oportunamente, para que pueda unirse con anticipación. Debe probar el ingreso al servidor de Discord antes del examen.
- Para acceder al servidor de discord use el enlace <https://discord.gg/3trUJdtF3B>
- En este servidor de Discord, existirá un canal por cada pregunta del examen parcial. De esta manera, si por ejemplo tuviera dudas acerca del enunciado de la pregunta 1A, podrá acceder al canal #pregunta-1a, y escribir su consulta allí. Un profesor del curso responderá sus consultas por chat. No podrá conversar con los otros alumnos en este canal, solo podrá conversar con los profesores.
- No está permitido que durante el examen, esté conectado a otros canales en Discord que no sean los exclusivos para realizar consultas sobre las preguntas del examen. El incumplir esta indicación, será motivo de anulación del examen.

### INDICACIONES GENERALES:

- Debe utilizar variables y constantes descriptivas, comentarios que expliquen el funcionamiento de la solución y mensajes descriptivos.
- El orden y la eficiencia de su implementación serán considerados en la calificación.

---

# Pregunta 1 – Pseudocódigo con selectivas anidadas (7 puntos)

## Pregunta 1A (7 puntos) [propuesta por David Allasi]

Una circunferencia se define como el lugar geométrico de los puntos del plano equidistantes de otro, llamado centro de la circunferencia.

Los elementos de una circunferencia se pueden apreciar en la figura 1 y las vamos a nombrar a continuación:

- Centro: punto central que está a la misma distancia de todos los puntos pertenecientes a la circunferencia.
- Radio: Parte de recta que une el centro con cualquier punto perteneciente a la circunferencia.
- Diámetro: Mayor cuerda que une dos puntos de una circunferencia. Hay infinitos diámetros y todos pasan por el centro de la circunferencia.
- Arco: Porción de la circunferencia, que queda definido a partir de dos puntos sobre dicha circunferencia.

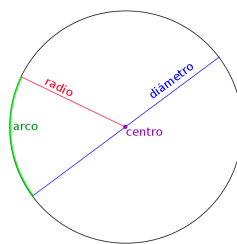


Figura 1: Elementos de la circunferencia

En la circunferencia se pueden formar varios tipos de ángulos, entre ellos podemos ver:

- **Ángulo central:** El ángulo central tiene su vértice en el centro de la circunferencia y sus lados son dos radios (ver figura 2).

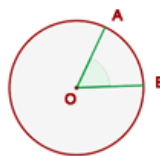


Figura 2: Ángulo central en la circunferencia

- **Ángulo inscrito:** El ángulo inscrito tiene su vértice en la circunferencia y sus lados son rectas secantes a ella (ver figura 3).
- **Ángulo semi-inscrito:** El vértice de ángulo semi-inscrito está en la circunferencia, un lado es una recta secante y el otro es una recta tangente a ella (ver figura 4).

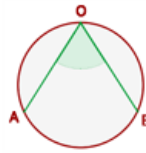


Figura 3: Ángulo inscrito en la circunferencia

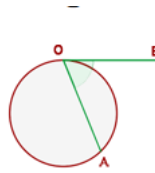


Figura 4: Ángulo semi-inscrito en la circunferencia

Además, la ecuación de la circunferencia con centro  $C(h, k)$  es  $(x - h)^2 + (y - k)^2 = r^2$ , donde  $r$  es el radio de la circunferencia.

Se le pide elaborar un algoritmo expresado en pseudocódigo que permita leer las coordenadas del centro de una circunferencia  $C(h, k)$ , donde  $h$  y  $k$  deben estar en el primer cuadrante del plano cartesiano, y su radio, el cual debe ser mayor que cero. Además, lea las coordenadas de dos puntos  $A(x_1, y_1)$  y  $B(x_2, y_2)$ , las coordenadas del punto  $O(x_3, y_3)$  que representa al vértice del ángulo y determine si los puntos ingresados y el vértice ingresado forman un ángulo central, inscrito o semi-inscrito o no se forma ninguno de los tipos de ángulos en la circunferencia. En caso de formar un ángulo central, se debe solicitar el ángulo AOB (en grados sexagesimales), debe validar que el ángulo ingresado sea mayor que cero, y debe calcular el perímetro y área del triángulo formado por los puntos A, O y B. En caso de formar un ángulo semi-inscrito, debe indicar que lado pertenece a una recta tangente y que lado pertenece a una recta secante, además debe calcular la distancia entre los puntos del lado que pertenece a la recta tangente.

En esta pregunta se deben mostrar mensajes específicos ante las siguientes situaciones:

- Si las coordenadas del centro de la circunferencia no están en el primer cuadrante, debe mostrar el mensaje `El centro de la circunferencia no pertenece al primer cuadrante y el algoritmo debe terminar.`
- Si el radio de la circunferencia no es mayor que cero, debe mostrar el mensaje `El radio de la circunferencia ingresado no es correcto y el algoritmo debe terminar.`
- Si el ángulo AOB (en grados sexagesimales) no es mayor que cero, debe mostrar el mensaje `El ángulo AOB ingresado no es correcto y el algoritmo debe terminar.`
- Si no se logra formar alguno de los ángulos descritos, se debe mostrar el siguiente mensaje `El vértice O y los puntos A y B no forman un ángulo descrito.`

### Comparación de números reales

Muchas veces el resultado de la comparación de números reales a través de la igualdad no es el deseado. En este caso es recomendable usar el valor absoluto de la diferencia de los números que se desean comparar. Si esta diferencia es cercana a cero (en esta ocasión será menor a 0.01), se puede asumir que son iguales. Considere que en PSeInt se utiliza la función `abs()` para obtener el valor absoluto de un número.

## Área del triángulo

Área triángulo AOB =  $\frac{LadoOA * LadoOB * Sen(anguloAOB)}{2}$ . En PSeInt existe una función sen para calcular el valor del seno de un ángulo en radianes.

## Distancia entre dos puntos

La distancia entre dos puntos  $P(x_1, y_1)$  y  $Q(x_2, y_2)$  se calcula con la siguiente fórmula:  
 $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

A continuación se presentan algunos ejemplos de ejecución del algoritmo: utilice los mensajes que se muestran dentro del desarrollo del pseudocódigo.

```
Ingrese las coordenadas del centro de la circunferencia:  
> 2  
> -1  
El centro de la circunferencia no pertenece al primer cuadrante
```

```
Ingrese las coordenadas del centro de la circunferencia:  
> 2  
> 3  
Ingrese el radio:  
> 0  
El radio de la circunferencia ingresado no es correcto
```

```
Ingrese las coordenadas del centro de la circunferencia:  
> 5.97  
> 3.44  
Ingrese el radio:  
> 2  
Ingrese las coordenadas del punto A:  
> 4.419  
> 2.179  
Ingrese las coordenadas del punto B:  
> 5.369  
> 5.349  
Ingrese las coordenadas del punto O:  
> 5.97  
> 3.44  
El vértice O y los puntos A y B forman un ángulo central.  
Ingrese el ángulo central AOB (en grados sexagesimales):  
> 50  
Datos del triángulo formado por AOB:  
Perímetro: 7.3092899541  
Área: 3.0641773057
```

```
Ingrese las coordenadas del centro de la circunferencia:  
> 5.97  
> 3.44  
Ingrese el radio:  
> 2  
Ingrese las coordenadas del punto A:  
> 4.419  
> 2.179  
Ingrese las coordenadas del punto B:  
> 5.369  
> 5.349  
Ingrese las coordenadas del punto O:
```

```

> 5.97
> 3.44
El vértice O y los puntos A y B forman un ángulo central.
Ingrese el ángulo central AOB (en grados sexagesimales):
> 0
El ángulo AOB ingresado no es correcto.

```

```

Ingrese las coordenadas del centro de la circunferencia:
> 5.97
> 3.44
Ingrese el radio:
> 2
Ingrese las coordenadas del punto A:
> 4.419
> 2.179
Ingrese las coordenadas del punto B:
> 7.77
> 2.563
Ingrese las coordenadas del punto O:
> 5.369
> 5.349
El vértice O y los puntos A y B forman un ángulo inscrito.

```

```

Ingrese las coordenadas del centro de la circunferencia:
> 5.97
> 3.44
Ingrese el radio:
> 2
Ingrese las coordenadas del punto A:
> 4.419
> 2.179
Ingrese las coordenadas del punto B:
> 7.77
> 2.563
Ingrese las coordenadas del punto O:
> 3
> 2
El vértice O y los puntos A y B no forman un tipo de ángulo descrito.

```

```

Ingrese las coordenadas del centro de la circunferencia:
> 5.97
> 3.44
Ingrese el radio:
> 2
Ingrese las coordenadas del punto A:
> 5.369
> 5.349
Ingrese las coordenadas del punto B:
> 9.7
> 2.01
Ingrese las coordenadas del punto O:
> 6.29
> 1.465
El vértice O y los puntos A y B forman un ángulo semi-inscrito.
El lado OB formado pertenece a la recta tangente y el lado OA pertenece a la recta secante.
La distancia del lado OB que pertenece a la recta tangente es: 3.4532774288.

```

### Programa 1: Propuesta de solución

```

1  Algoritmo PreguntalA
2  Escribir "Ingrese las coordenadas del centro de la circunferencia: "
3  Leer h, k

```

```

4 Si ((h>0) y (k>0)) Entonces
5   Escribir "Ingrese el radio: "
6   Leer radio
7   Si (radio>0) entonces
8     Escribir "Ingrese las coordenadas del punto A: "
9     Leer x1,y1
10    Escribir "Ingrese las coordenadas del punto B: "
11    Leer x2,y2
12    Escribir "Ingrese las coordenadas del punto O: "
13    Leer x3,y3
14    pertenece_A <- abs((x1-h)^2 + (y1-k)^2 - radio^2) < 0.01
15    pertenece_B <- abs((x2-h)^2 + (y2-k)^2 - radio^2) < 0.01
16    pertenece_O <- abs((x3-h)^2 + (y3-k)^2 - radio^2) < 0.01
17    Si pertenece_A y pertenece_B entonces
18      Si pertenece_O Entonces
19        Escribir "El vértice O y los puntos A y B forman un ángulo inscrito"
20      SiNo
21        Si (abs(x3-h)<0.01 y abs(y3-k)<0.01) Entonces
22          Escribir "El vértice O y los puntos A y B forman un ángulo central"
23          distanciaAB <- rc((x2-x1)^2 + (y2-y1)^2)
24          perimetro <- distanciaAB + 2*radio
25          Escribir "Ingrese el ángulo central AOB (en grados sexagesimales): "
26          Leer angulo
27          Si (angulo>0) entonces
28            area <- radio*radio*sen(angulo*3.141592/180)
29            Escribir "Datos del triángulo formado por AOB: "
30            Escribir "Perímetro: ", perimetro
31            Escribir "Área: ", area
32          SiNo
33            Escribir "El ángulo AOB ingresado no es correcto."
34          Fin Si
35        SiNo
36          Escribir "El vértice O y los puntos A y B no forman un ángulo descrito"
37        FinSi
38      FinSi
39    SiNo
40      Si (pertenece_A y no pertenece_B) Entonces
41        Si (pertenece_O) Entonces
42          Escribir "El vértice O y los puntos A y B forman un ángulo semi-inscrito"
43          Escribir "El lado OB formado pertenece a la recta tangente y el lado OA pertenece a la recta secante"
44          distanciaOB <- rc((x3-x2)^2 + (y3-y2)^2)
45          Escribir "La distancia del lado OB que pertenece a la recta tangente es: ", distanciaOB
46        SiNo
47          Escribir "El vértice O y los puntos A y B no forman un ángulo descrito"
48        FinSi
49      SiNo
50        Si (no pertenece_A y pertenece_B) Entonces
51          Si (pertenece_O) Entonces
52            Escribir "El vértice O y los puntos A y B forman un ángulo semi-inscrito"
53            Escribir "El lado OB formado pertenece a la recta secante y el lado OA pertenece a la recta tangente"
54            distanciaOA <- rc((x3-x1)^2 + (y3-y1)^2)
55            Escribir "La distancia del lado OA que pertenece a la recta tangente es: ", distanciaOA
56          SiNo
57            Escribir "El vértice O y los puntos A y B no forman un ángulo descrito"
58          FinSi
59        SiNo
60          Escribir "El vértice O y los puntos A y B no forman un ángulo descrito"
61        FinSi
62      FinSi
63    Fin Si
64  SiNo
65    Escribir "El radio de la circunferencia ingresado no es correcto"
66  Fin Si
67 SiNo
68   Escribir "El centro de la circunferencia no pertenece al primer cuadrante"
69 FinSi
70 FinAlgoritmo

```



**Pregunta 1B (7 puntos)** [propuesta por Silia Vargas]

En la figura 5 se muestra un plano inclinado con un bloque, el movimiento en este escenario depende si existe fricción o no entre el bloque y el plano.

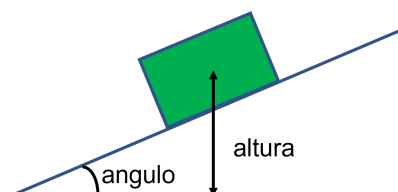


Figura 5: Movimiento de un bloque en un plano inclinado

En las tablas, tabla 1 "Fórmulas en un plano inclinado sin fricción" y tabla 2 "Fórmulas en un plano inclinado con fricción" se muestran las fórmulas para ambos casos.

Tabla 1: Fórmulas en un plano inclinado sin fricción

Fórmulas en un plano inclinado sin fricción
$aceleracion = gravedad \times sen(angulo)$
$tiempo = \frac{1}{sen(angulo)} \sqrt{\frac{2 \times altura}{gravedad}}$
$velocidad = \sqrt{2 \times gravedad \times altura}$

Tabla 2: Fórmulas en un plano inclinado con fricción

Fórmulas en un plano inclinado con fricción. El coeficiente corresponde al coeficiente de rozamiento dinámico.
$aceleracion = gravedad \times sen(angulo) - coeficiente \times gravedad \times cos(angulo)$
$tiempo = \frac{1}{\sqrt{sen(angulo) \times (sen(angulo) - coeficiente \times cos(angulo))}} \times \sqrt{\frac{2 \times altura}{gravedad}}$
$velocidad = \sqrt{\frac{sen(angulo) - coeficiente \times cos(angulo)}{sen(angulo)}} \times \sqrt{2 \times gravedad \times altura}$

**Se pide elaborar un algoritmo expresado en pseudocódigo** que permita leer la altura junto con su unidad de medida; luego debe leer el coseno del ángulo que forma el plano inclinado con el piso; posteriormente debe solicitar si existe o no fricción en el plano inclinado; y con ello calcular la aceleración, el tiempo y la velocidad del movimiento del bloque.

Cada dato que lea debe validarse. Para la altura se debe ingresar un valor positivo y las unidades de medida válidas son centímetros c, decímetros d o metros m. Si los datos de la altura no son válidos, debe mostrar el mensaje

”Los datos de la altura no son correctos”. El coseno del ángulo debe estar dentro del rango [-1,1]. Si el coseno no es válido debe mostrar el mensaje ” El coseno del ángulo no es correcto”.

Para indicar que hay fricción se puede ingresar s o S y si no hay fricción n o N. Si no se ingresa correctamente esta información se debe mostrar el mensaje ”No se ingresó correctamente si existe o no fricción entre el bloque y el plano inclinado”. En caso esta información sea correcta, debe calcular el ángulo entre el plano inclinado y el piso, validar que el ángulo sea mayor que 0 y menor que 90 grados sexagesimales; si el ángulo calculado no es válido debe mostrar el mensaje ”El ángulo debe ser mayor a 0 y menor a 90 grados sexagesimales”. Si el valor del ángulo es válido, realizar y mostrar los cálculos respectivos para el bloque en el plano inclinado. En caso exista fricción debe solicitar el coeficiente de rozamiento dinámico, debe validar que sea positivo y menor o igual que la tangente del ángulo calculado. En caso el coeficiente no cumpla con esta validación debe mostrar el mensaje ”El coeficiente debe ser positivo o no existe movimiento porque el coeficiente es mayor a la tangente del ángulo”.

Debe mostrar los resultados en metros, cuando exista fricción use minutos y cuando no exista fricción use segundos.

Recuerde que:

- $\text{angulo} = \arccos(\text{coseno})$
- $\text{gravedad} = 9.81 \text{ m/s}^2$
- $360^\circ = 2\pi$
- 1 metro = 100 centímetros
- 1 metro = 10 decímetros
- $1 \text{ minuto}^2 = 3600 \text{ segundos}^2$

A continuación se presentan algunos ejemplos de ejecución del programa: utilice los mensajes que se muestran dentro del desarrollo del pseudocódigo.

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 50
> c
Ingrese el coseno del ángulo:
> 0.5
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> n
Cuando NO existe fricción entre el objeto y el plano inclinado:
La aceleracion es 8.4957092111 m/s2
El tiempo que tarda el bloque en llegar al final del plano incli nado es: 0.3686675091 s
La velocidad final del bloque es 3.1320919527 m/s
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 5
> d
Ingrese el coseno del ángulo:
> 0.5
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> s
Ingrese el coeficiente de rozamiento dinámico:
> 1
Cuando Sí existe fricción entre el objeto y el plano inclinado:
La aceleracion es 0.0009974192 m/min2
El tiempo que tarda el bloque en llegar al final del plano inclinado es: 0.0094513385 min
La velocidad final del bloque es 122.1732290369 m/min
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 5
> m
Ingrese el coseno del ángulo:
> 0.7
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> S
Ingrese el coeficiente de rozamiento dinámico:
> 0.99
Cuando SÍ existe fricción entre el objeto y el plano inclinado:
La aceleracion es 0.0000576142 m/min2
El tiempo que tarda el bloque en llegar al final del plano inclinado es: 0.1369430546 min
La velocidad final del bloque es 102.2527274384 m/min
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 10
> c
Ingrese el coseno del ángulo:
> 0.67
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> n
Cuando NO existe fricción entre el objeto y el plano inclinado:
La aceleracion es 7.2825623725 m/s2
El tiempo que tarda el bloque en llegar al final del plano inclinado es: 0.1923380854 s
La velocidad final del bloque es 1.4007141036 m/s
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 10
> M
Los datos de la altura no son correctos
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> -2.6
> m
Los datos de la altura no son correctos
```

```
Ingrese la altura del bloque al piso y su unidad de medida(c:centímetros, d: decímetros,
m: metros):
> 50
> d
Ingrese el coseno del ángulo:
> 3.4
El coseno del ángulo no es correcto
```

```
Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 10
> c
Ingrese el coseno del ángulo:
> 0.75
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> M
No se ingresó correctamente si existe o no fricción entre el bloque y el plano inclinado
```

```

Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 40
> d
Ingrese el coseno del ángulo:
> 0.45
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> S
Ingrese el coeficiente de rozamiento dinámico:
> -3
El coeficiente debe ser positivo o no existe movimiento porque el coeficiente es mayor a la
tangente del ángulo

```

```

Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros,
m: metros):
> 10
> c
Ingrese el coseno del ángulo:
> 0.67
Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n):
> S
Ingrese el coeficiente de rozamiento dinámico:
> 5
El coeficiente debe ser positivo o no existe movimiento porque el coeficiente es mayor a la
tangente del ángulo

```

## Programa 2: Propuesta de solución

```

1  Algoritmo PlanoInclinadoconRozamiento
2  Escribir "Ingrese la altura del bloque al piso y su unidad de medida (c:centímetros, d: decímetros, m: metros): "
3  Leer altura, unidadAltura
4  alturaValida<-altura>0 Y (unidadAltura='c' O unidadAltura='d' O unidadAltura='m')
5  Si alturaValida entonces
6      Escribir "Ingrese el coseno del ángulo:"
7      Leer cosAngulo
8      Si cosAngulo>=1 Y cosAngulo<=1 Entonces
9          Escribir "Ingrese si existe fricción entre el bloque y el plano inclinado, sí (S o s), no (N o n): "
10         Leer friccion
11         friccionValida<-friccion='S' O friccion=='s' o friccion='N' o friccion='n'
12         Si friccionValida Entonces
13             angulo<-acos(cosAngulo)
14             anguloSex<-angulo*180/PI
15             GRAVEDAD<-9.81
16             Si anguloSex>0 Y anguloSex<90 Entonces
17                 Si unidadAltura='c' Entonces
18                     altura<-altura/100
19                 SiNo
20                     Si unidadAltura='d' Entonces
21                         altura<-altura/10
22                     FinSi
23                 FinSi
24                 Si friccion='n' O friccion='N' Entonces
25                     aceleracion<-GRAVEDAD*sen(angulo)
26                     tiempo<-(1/sen(angulo))*rc(2*altura/GRAVEDAD)
27                     velocidad<-rc(2*GRAVEDAD*altura)
28                     Escribir "Cuando NO existe fricción entre el objeto y el plano inclinado:"
29                     Escribir "La aceleracion es ", aceleracion, " m/s2"
30                     Escribir "El tiempo que tarda el bloque en llegar al final del plano inclinado es: ", tiempo, " s"
31                     Escribir "La velocidad final del bloque es ", velocidad " m/s"
32                 SiNo
33                     Escribir "Ingrese el coeficiente de rozamiento dinámico: "
34                     Leer coeficiente
35                     tangAngulo<-tan(angulo)
36                     Si coeficiente>0 Y coeficiente<=tangAngulo entonces

```

```

37         senAngulo<-sen(angulo)
38         aceleracion<-(GRAVEDAD*senAngulo-coeficiente*GRAVEDAD*cosAngulo)/3600
39         tiempo<-((1/rc(senAngulo*(senAngulo-coeficiente*cosAngulo)))*rc(2*altura/GRAVEDAD))/60
40         velocidad<-rc((senAngulo-coeficiente*cosAngulo)/senAngulo)*rc(2*GRAVEDAD*altura)*60
41         Escribir "Cuando SÍ existe fricción entre el objeto y el plano inclinado:"
42         Escribir "La aceleracion es ", aceleracion, " m/min2"
43         Escribir "El tiempo que tarda el bloque en llegar al final del plano inclinado es: ", tiempo, " min"
44         Escribir "La velocidad final del bloque es ", velocidad " m/min"
45     SiNo
46         Escribir "El coeficiente debe ser positivo o no existe movimiento porque el coeficiente es mayor a la tangente del ángulo"
47     FinSi
48     FinSi
49     SiNo
50         Escribir "El ángulo debe ser mayor a 0 y menor a 90 grados sexagesimales"
51     FinSi
52     SiNo
53         Escribir "No se ingresó correctamente si existe o no fricción entre el bloque y el plano inclinado"
54     FinSi
55     SiNo
56         Escribir "El coseno del ángulo no es correcto"
57     FinSi
58
59     SiNo
60         Escribir "Los datos de la altura no son correctos"
61     FinSi
62 FinAlgoritmo

```

El archivo debe ser grabado con el nombre **E1\_código\_pregunta1.psc** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta1.psc**, sin espacios).

# Pregunta 2 – Programa en C con selectivas anidadas y programación modular (7 puntos)

## Pregunta 2A (7 puntos) [propuesta por Layla Hirsh]

La información personal es cualquier dato relacionado a una persona, ya sea que se refiera a la vida privada o pública. En el entorno en línea, donde una vasta cantidad de información es compartida y transferida alrededor del mundo de forma inmediata, se hace cada vez más difícil para las personas mantener un control sobre su información personal. Aquí es donde la protección de datos entra en juego.

La protección de datos se refiere a las prácticas, salvaguardas y principios fundamentales puestos en ejercicio para proteger la información personal y asegurar que se mantenga en control de ella. En pocas palabras, uno debe tener la posibilidad de decidir si se desea o no compartir ciertos datos, quién puede tener acceso a ellos, por cuánto tiempo, por qué razones, tener la posibilidad de modificarlos y mucho más.

Los gobiernos tienen un interés de seguridad en garantizar la protección de datos personales. En 2015, un grupo de delincuentes robaron una cantidad de 21,5 millones de registros de la Oficina de Administración de Personal de EE. UU. que contenían información personal sumamente sensible de los empleados federales y de los miembros de su familia. Este tipo de ataques suceden cada vez con mayor frecuencia en todo el mundo, por lo tanto, los países deben tomar medidas para proteger la información de las personas de forma eficiente.

Es por ello que la encriptación de datos es tan importante al momento de querer almacenar datos en una base de datos. Juan, alumno de ingeniería sabe esto y es por ello que quiere implementar un programa en C que le permita generar de manera segura un identificador encriptado para cada uno de los participantes en un estudio que va a realizar.

A continuación se describen las partes del identificador no encriptado:

- Inicial de nombre: dos primeros dígitos de más a la izquierda que representan una letra entre la A y la Z (código ASCII mayúsculas).
- Inicial de apellido Paterno: tercer y cuarto dígito contando desde la izquierda que representan una letra entre la A y la Z (código ASCII mayúsculas).
- Inicial de apellido Materno: quinto y sexto dígito contando desde la izquierda que representan una letra entre la A y la Z (código ASCII mayúsculas).
- Documento nacional de identidad: los ocho dígitos de más a la derecha.

Según lo definido por Juan, existen tres tipos de encriptación:

- **A:** formada por Documento nacional de identidad, seguido por la inicial de apellido materno, seguido por la inicial de apellido paterno y finalmente la inicial del nombre.
- **B:** formada por la inicial del apellido materno, seguido por la inicial del apellido paterno, seguido de la inicial del nombre y finalmente el Documento nacional de identidad.
- **C:** formada por la inicial del nombre, seguido del Documento nacional de identidad, seguido de la inicial del apellido materno, y finalmente la inicial del apellido paterno.

**Se le pide elaborar un programa en lenguaje C**, que dado el identificador de un participante y un modo de encriptación, permita mostrar las iniciales del nombre, apellidos paterno y materno además del documento nacional de identidad y calcular y mostrar el identificador encriptado siempre y cuando los datos ingresados sean válidos.

Para esto debe considerar que el identificador es un número de 14 dígitos para el desarrollo de esta pregunta, no puede usar estructuras iterativas y debe tomar en cuenta que el tipo de dato int tiene un valor máximo de 2147483647. Recuerde que solo existen tres modos de encriptación A, B, C (ingresados en mayúsculas), en caso el usuario ingrese un modo inválido debe mostrar el mensaje "Modo inválido". Por otro lado, si el identificador no tiene los 14 dígitos requeridos deberá mostrar el mensaje "El identificador no cumple con los requerimientos". Considere también que si alguna de las iniciales no es válida, deberá mostrar el mensaje "Iniciales inválidas".

Para resolver este problema, deberá implementar cuatro módulos incluido el módulo principal, según el siguiente detalle:

- Un módulo que permita verificar que tanto el modo como el identificador cumplen las condiciones dadas.
- Un módulo para validar que las iniciales del nombre y las iniciales de los dos apellidos sean válidas.
- Un módulo que reciba como parámetros el identificador original, y calcule y devuelva el código ASCII de la inicial del nombre, el código ASCII de la inicial del apellido paterno, el código ASCII de la inicial del apellido materno y el número del documento nacional de identidad.

#### Tomar en cuenta:

- No deberá implementar otros módulos diferentes a los indicados previamente.
- No deberá usar sentencias iterativas en esta pregunta.

#### Casos de prueba para verificar la solución:

```
Ingrese el identificador del participante: 76807742745364
Identificador leído:76807742745364
Ingrese el modo de encriptación: A
Inicial nombre:L
Inicial apellido paterno:P
Inicial apellido materno:M
DNI:42745364
Usando el modo de encriptación A
El identificador encriptado es:42745364778076
```

```
Ingrese el identificador del participante: 82748775647382
Identificador leído:82748775647382
Ingrese el modo de encriptación: C
Inicial nombre:R
Inicial apellido paterno:J
Inicial apellido materno:W
DNI:75647382
Usando el modo de encriptación C
El identificador encriptado es:82756473828774
```

```
Ingrese el identificador del participante: 76807742745364
Identificador leído:76807742745364
Ingrese el modo de encriptación: B

Inicial nombre:L
Inicial apellido paterno:P
Inicial apellido materno:M
DNI:42745364
Usando el modo de encriptación B
El identificador encriptado es:77807642745364
```

```
Ingrese el identificador del participante: 98656675453243
Identificador leído:98656675453243
Ingrese el modo de encriptación: A
```

```
Inicial nombre:b
Inicial apellido paterno:A
Inicial apellido materno:B
DNI:75453243
Usando el modo de encriptación A
Iniciales inválidas
```

```
Ingrese el identificador del participante: 65666787654565
Identificador leído:65666787654565
Ingrese el modo de encriptación: P
Modo inválido
```

```
Ingrese el identificador del participante: 76489
Identificador leído:76489
Ingrese el modo de encriptación: C
El identificador no cumple con los requerimientos
```

```
Ingrese el identificador del participante: 43257
Identificador leído:43257
Ingrese el modo de encriptación: V
```

### Programa 3: Propuesta de solución

```
1  #include<stdio.h>
2  #define DIEZ_6 1000000
3  #define DIEZ_4 10000
4  #define DIEZ_2 100
5  #define DIEZ_10 10000000000
6  #define DIEZ_8 100000000
7  #define DIEZ_12 1000000000000
8  #define DIEZ_14 100000000000000
9  #define DIEZ_13 10000000000000
10
11 int ValidarDatosEntrada(double id,char modo);
12 int InicialesValidas(int nombre,int apellPaterno,int apellMaterno);
13 void extraePartes(double id,int *nombre,int *apellPaterno,int *apellMaterno,double *dni);
14 int main(){
15     double nuevo,id,dni,suma;
16     int nombre,apellPaterno,apellMaterno,sonValidas;
17     char modo;
18     printf("Ingrese el identificador del participante: ");
19     scanf("%lf",&id);
20     printf("Identificador leído: %.0lf",id);
21     printf("\nIngrese el modo de encriptación: ");
22     scanf("\n%c",&modo);
23     datosValidos=ValidarDatosEntrada(id,modo);
24     if(datosValidos){
25         extraePartes(id,&nombre,&apellPaterno,&apellMaterno,&dni);
26
27         printf("\nInicial nombre: %c",nombre);
28         printf("\nInicial apellido paterno: %c",apellPaterno);
29         printf("\nInicial apellido materno: %c",apellMaterno);
30         printf("\nDNI: %.0lf",dni);
31         printf("\nUsando el modo de encriptación %c",modo);
32         sonValidas=InicialesValidas(nombre,apellPaterno,apellMaterno);
33         if(sonValidas){
34             if (modo=='A'){
```



```

35 //DMPN
36 nuevo=dni*DIEZ_6+apellMaterno*DIEZ_4+apellPaterno*DIEZ_2+nombre;
37
38 }
39 else{
40     if (modo=='B'){
41         //MPND
42         nuevo=(double)apellMaterno*DIEZ_12+(double)apellPaterno*DIEZ_10+(double)nombre*
43             DIEZ_8+dni;
44     }
45     else{
46         //NDMP
47         nuevo=(double)nombre*DIEZ_12+dni*DIEZ_4+(double)apellMaterno*DIEZ_2+(double)
48             apellPaterno;
49     }
50     printf("\nEl identificador encriptado es: %.0lf",nuevo);
51 }
52 else{
53     printf("\n Iniciales inválidas");
54 }
55 }
56 else{
57     if(modo<'A' || modo>'C' ){
58         printf("Modo inválido");
59     }
60     else{
61         printf("El identificador no cumple con los requerimientos");
62     }
63 }
64 return 0;
65 }
66 int ValidarDatosEntrada(double id,char modo){
67     return modo>='A' && modo<='C' && id<DIEZ_14 && id >=DIEZ_13;
68 }
69 }
70 void extraePartes(double id,int *nombre,int *apellPaterno,int *apellMaterno,double *dni){
71     (*nombre)=(int)(id/DIEZ_12);
72     (*apellPaterno)=(int)((id-(*nombre)*DIEZ_12)/DIEZ_10);
73     (*apellMaterno)=(int)((id-(*nombre)*DIEZ_12-(*apellPaterno)*DIEZ_10)/DIEZ_8);
74     (*dni)=id-(double)(*nombre)*DIEZ_12-(double)(*apellPaterno)*DIEZ_10-(double)(*apellMaterno)*DIEZ_8;
75 }
76 int InicialesValidas(int nombre,int apellPaterno,int apellMaterno){
77     if(nombre>=(int)'A' && apellPaterno>=(int)'A' && apellMaterno>=(int)'A' &&
78         nombre<=(int)'Z' && apellPaterno<=(int)'Z' && apellMaterno<=(int)'Z' )
79         return 1;
80     else
81         return 0;
82 }

```

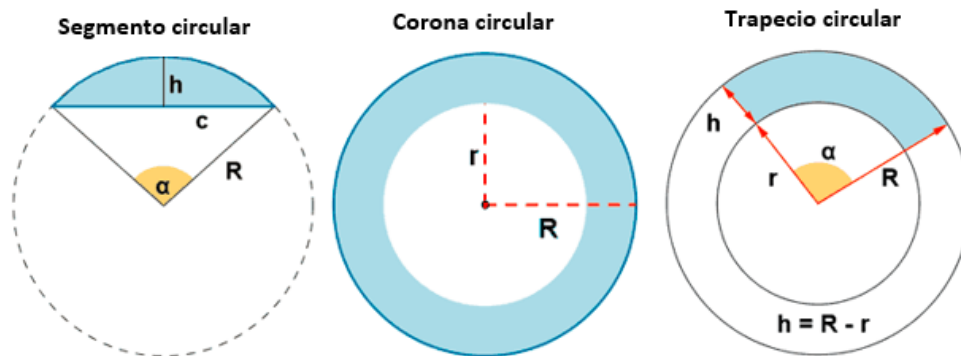
### **Pregunta 2B (7 puntos)** [propuesta por Jennifer Zárate]

El círculo es la región del plano delimitada por una circunferencia, la cual es una figura geométrica plana que se caracteriza porque todos sus puntos equidistan de un punto llamado centro. A continuación se describen algunas de las partes que tiene un círculo:

- Radio: es un segmento que tiene como extremos cualquier punto del borde de la circunferencia y el centro.
- Diámetro: es un segmento que tiene como extremos dos puntos del borde de la circunferencia y que siempre pasa por el centro. En términos de valores, es del doble de medida del radio.
- Cuerda: es un segmento que tiene como extremos dos puntos del borde de la circunferencia

- **Altura de cuerda:** es el segmento que une el punto central de la cuerda con el punto de la circunferencia que forman una perpendicular.

De la circunferencia se desprenden distintas figuras circulares y son usadas en diseños de todo tipo, como por ejemplo, para la construcción de rotondas, diseño geométrico de carreteras, construcción de tanques o reservorios, entre otros. A continuación, se muestran algunos tipos:



- **Segmento circular:** es una parte del círculo que está comprendida entre la cuerda y el arco.
- **Corona circular:** es una figura geométrica delimitada por dos circunferencias que tienen el mismo centro pero radios diferentes.
- **Trapecio circular:** es una sección de una corona circular delimitada por un ángulo establecido.

Se le pide elaborar un programa en lenguaje C, que dado un tipo de figura a evaluar identificado con la letra 'C' y número asociado a una figura circular, permita calcular su longitud o área según se especifique. Para ello, deberá solicitar los valores necesarios para realizar los cálculos dependiendo del número elegido. El menú de opciones para especificar qué figura circular se elegirá es: (1) Segmento circular, (2) Corona Circular y (3) Trapecio Circular. Asimismo, las opciones para seleccionar el cálculo a realizar es la letra 'L' para la longitud o la letra 'A' para el área. En la tabla 3 "Fórmulas Figuras Circulares" se muestran las fórmulas que deberá utilizar.

Para resolver este problema, deberá implementar tres módulos sin incluir el módulo principal, según el siguiente detalle:

- Un módulo para validar el tipo de figura a evaluar y las opciones del menú de figuras geométricas.
- Un módulo que reciba como parámetros el radio, el ángulo  $\alpha$ , la longitud de cuerda y la altura  $h$ , y que calcule y devuelva el área del segmento circular.
- Un módulo que reciba como parámetros el radio mayor y radio menor, que calcule y devuelva el área de la corona circular y el área del círculo mayor.

#### Tomar en cuenta:

- **No deberá implementar otros módulos diferentes a los indicados previamente.**
- **No deberá usar sentencias iterativas en esta pregunta.**

#### Casos de prueba para verificar la solución:

```

Ingrese el tipo de figura a evaluar (C)Círculo: C
Ingrese la opción de la figura circular a analizar (1)Segmento circular   (2)Corona
Circular   (3)Trapecio Circular: 3
Ingrese el resultado que desea calcular (L)Longitud   (A)Área: A
Ingrese los valores de alfa, radio mayor y radio menor: : 180 10 5
El área del trapecio circular es: 117.81
  
```

Tabla 3: Fórmulas Figuras Circulares

Fórmulas:	
<b>Segmento Circular</b>	$longitud = 0.01745R\alpha + c$ $Area = \frac{\pi R^2 \alpha}{360} - \frac{c(R-h)}{2}$
<b>Corona Circular</b>	$longitud = 2\pi(R+r)$ $Area = \pi(R^2 - r^2)$
<b>Trapezio Circular</b>	$longitud = \frac{2\pi(R+r)\alpha}{360} + 2(R-r)$ $Area = \frac{\pi(R^2 - r^2)\alpha}{360}$
donde:	<ul style="list-style-type: none"> <li>▪ <math>R</math>: radio de circunferencia mayor</li> <li>▪ <math>r</math>: radio de circunferencia menor</li> <li>▪ <math>\alpha</math>: ángulo en sexagesimales</li> <li>▪ <math>c</math>: longitud de cuerda</li> <li>▪ <math>h</math>: altura del segmento circular</li> <li>▪ Considerar el valor de <math>\pi=3.14159</math></li> </ul>

Ingrese el tipo de figura a evaluar (C)Círculo: C  
 Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: 4  
 Ingresó una opción inválida para identificar a la figura circular.

Ingrese el tipo de figura a evaluar (C)Círculo: C  
 Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: 1  
 Ingrese el resultado que desea calcular (L)Longitud (A)Área: C  
 Ingresó una opción inválida para el cálculo a realizar.

Ingrese el tipo de figura a evaluar (C)Círculo: C  
 Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: 2  
 Ingrese el resultado que desea calcular (L)Longitud (A)Área: A  
 Ingrese los valores de radio mayor y radio menor: 20 15  
 El área de la corona circular es 549.78 y el área del círculo es 1256.64

```

Ingrese el tipo de figura a evaluar (C)Círculo: C
Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: 1
Ingrese el resultado que desea calcular (L)Longitud (A)Área: L
Ingrese los valores de radio, cuerda, alfa, altura: 30 10 90 8
La longitud del segmento circular es: 57.11

```

```

Ingrese el tipo de figura a evaluar (C)Círculo: A
Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: 3
Ingresó una opción inválida para identificar a la figura circular.

```

#### Programa 4: Propuesta de solución

```

1  #include<stdio.h>
2  #include<math.h>
3  #define PI 3.14159
4
5  int ValidarOpcionMenu(char tipoFigura,int opcion);
6  double CalcularAreaSegmentoCircular(double radio,double cuerda,double alfa,double h);
7  void CalcularAreasCoronaCircularYCirculo(double radioMayor,double radioMenor,double *areaCorona,double *areaCirculo);
8
9  int main(){
10     char tipoFigura,opcion;
11     int opcionValida,opcion;
12     double radio,cuerda,alfa,h,longitud,radioMayor,radioMenor,area,areaCorona,areaCirculo;
13     printf("Ingrese el tipo de figura a evaluar (C)Círculo: ");
14     scanf("%c",&tipoFigura);
15     printf("Ingrese la opción de la figura circular a analizar (1)Segmento circular (2)Corona Circular (3)Trapezio Circular: ");
16     scanf("%d",&opcion);
17     opcionValida=ValidarOpcionMenu(tipoFigura,opcion); /*validar la opción del menú ingresada*/
18     if (opcionValida){ /*solo si la opción es válida, se pide ingresar la operación a realizar*/
19         printf("Ingrese el resultado que desea calcular (L)Longitud (A)Área: ");
20         scanf("\n%c",&opcion);
21         if (opcion=='L' || operacion=='A'){
22             if (opcion==1){/*Se trata de un segmento circular*/
23                 printf("Ingrese los valores de radio, cuerda, alfa, altura: ");
24                 scanf("%lf %lf %lf %lf",&radio,&cuerda,&alfa,&h);
25                 if (opcion=='L'){
26                     longitud= 0.01745*radio*alfa+cuerda;
27                     printf("La longitud del segmento circular es: %.2lf",longitud);
28                 }
29                 else{
30                     area=CalcularAreaSegmentoCircular(radio,cuerda,alfa,h);
31                     printf("El área del segmento circular es: %.2lf",area);
32                 }
33             }
34             else if (opcion==2){/*Se trata de una corona circular – también deberá hallar el área del círculo*/
35                 printf("Ingrese los valores de radio mayor y radio menor: ");
36                 scanf("%lf %lf",&radioMayor,&radioMenor);
37                 if (opcion=='L'){
38                     longitud= 2*PI*(radioMayor+radioMenor);
39                     printf("La longitud de la corona circular es: %.2lf",longitud);
40                 }
41                 else{
42                     CalcularAreasCoronaCircularYCirculo(radioMayor,radioMenor,&areaCorona,&areaCirculo);
43                     printf("El área de la corona circular es %.2lf y el área del círculo es %.2lf",areaCorona,areaCirculo);
44                 }
45             }
46             else if (opcion==3){ /*Se trata de trapezio circular*/
47                 printf("Ingrese los valores de alfa, radio mayor y radio menor: : ");
48                 scanf("%lf %lf %lf",&alfa,&radioMayor,&radioMenor);

```

```

49         if (operacion=='L'){
50             longitud= 2*PI*(radioMayor+radioMenor)*alfa/360 + 2*(radioMayor-radioMenor);
51             printf("La longitud del trapecio circular es: %.2lf",longitud);
52         }
53         else{
54             area=PI*(pow(radioMayor,2)- pow(radioMenor,2))*alfa/360;
55             printf("El área del trapecio circular es: %.2lf",area);
56         }
57     }
58 }
59 else{
60     printf("Ingresó una opción inválida para el cálculo a realizar.\n");
61 }
62 }
63 else{
64     printf("Ingresó una opción inválida para identificar a la figura circular.\n");
65 }
66 return 0;
67 }
68
69 int ValidarOpcionMenu(char tipoFigura,int opcion){
70     return ((tipoFigura=='C') && ((opcion==1) || (opcion==2) || (opcion==3)));
71 }
72 double CalcularAreaSegmentoCircular(double radio,double cuerda,double alfa,double h){
73     return (PI*pow(radio,2)*alfa/360-cuerda*(radio-h)/2);
74 }
75 void CalcularAreasCoronaCircularYCirculo(double radioMayor,double radioMenor,double *areaCorona,double *areaCirculo){
76     *areaCorona=PI*(pow(radioMayor,2)-pow(radioMenor,2));
77     *areaCirculo=PI*(pow(radioMayor,2));
78 }

```

El archivo debe ser grabado con el nombre **E1\_código\_pregunta2.c** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta2.c**, sin espacios).

## Pregunta 3 – Programa en C con iterativas y programación modular (6 puntos)

### Pregunta 3A (6 puntos) *[propuesta por Sergio Ponce]*

Se cuenta con una serie de puntos en el plano cartesiano. **Se pide elaborar un programa en lenguaje C** que permita calcular si un punto se encuentra dentro del triángulo que forman otros 3 puntos. Para esto, el programa debe hacer lo siguiente:

- Leer la cantidad de puntos que el usuario debe ingresar y validar que dicho valor se encuentre entre 4 y 32; además, debe validar que el valor ingresado sea un múltiplo de 4. En caso el valor ingresado no cumpla con lo solicitado, deberá mostrar el mensaje “El valor ingresado no cumple con lo solicitado”.
- Para cada punto deberá leer el valor de la abscisa y de la ordenada. Asuma que ingresará los puntos en orden.
- Deberá validar si los tres primeros puntos ingresados en una iteración están alineados, si lo están, entonces no forman un triángulo e imprimirá “Los puntos están alineados, no forman un triángulo”.
- Por cada iteración se ingresarán los cuatro puntos uno después de otro, siendo el cuarto punto el que validaremos si se encuentra dentro del triángulo formado por los tres puntos anteriores.
- Al finalizar la ejecución, el programa debe mostrar la cantidad de puntos que se encontraban dentro y fuera de un triángulo.
- Para su solución debe considerar lo siguiente:
  - Un módulo que valide si los puntos dados están alineados. Debe recibir como parámetros la abscisa y ordenada de cada uno de los puntos que se utilizarán para la validación.
  - Un módulo para obtener la orientación de un triángulo. Debe recibir como parámetros la abscisa y ordenada de cada uno de los puntos que forman el triángulo.

### **Tomar en cuenta:**

- **No deberá implementar otros módulos diferentes a los indicados previamente.**

Recordar que:

Dado un triángulo ABC y un punto P del plano, P está en el interior del triángulo ABC (sin considerar las fronteras) si la orientación de los triángulos ABP, BCP y CAP es la misma orientación que la del triángulo ABC.

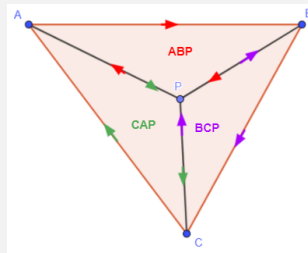


Figura 6: Si todos los triángulos formados con el punto P tienen la misma dirección que ABC, entonces P está dentro de ABC.

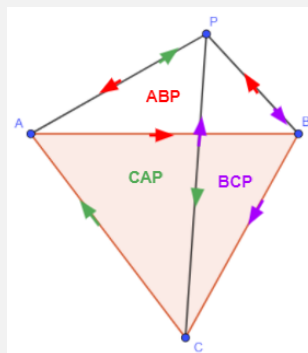


Figura 7: Si por lo menos uno de los triángulos formados con el punto P no tiene la misma dirección que ABC, entonces P está fuera de ABC.

Al calcular la orientación de un triángulo, si el resultado es mayor o igual a 0, la orientación será positiva, caso contrario la orientación será negativa.

Los pasos que debe tener en cuenta para su solución son los siguientes:

1. Calcular la orientación del triángulo ABC.
2. Calcular la orientación de los triángulos ABP, BCP y CAP.
3. Si la orientación del triángulo ABC es positiva y las orientaciones de los triángulo ABP, BCP y CAP también son positivas, entonces el punto P está dentro del triángulo ABC, caso contrario el punto P está fuera del triángulo ABC.
4. Si la orientación del triángulo ABC es negativa y las orientaciones de los triángulo ABP, BCP y CAP también son negativas, entonces el punto P está dentro del triángulo ABC, caso contrario el punto P está fuera del triángulo ABC.

Para calcular la orientación de un triángulo ABC cuyos puntos son  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$  debe utilizar la siguiente fórmula:

$$\text{orientacion} = (x_1 - x_3) * (y_2 - y_3) - (y_1 - y_3) * (x_2 - x_3)$$

**Recordar que:**

Dados tres puntos  $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3)$ , los tres puntos están alineados si los componentes de los vectores  $\vec{AB}$  y  $\vec{BC}$  son proporcionales

Tener en cuenta que:

$$\vec{AB} = (x_2 - x_1, y_2 - y_1)$$

$$\vec{BC} = (x_3 - x_2, y_3 - y_2)$$

Si los componentes son proporcionales, se cumplirá que:

$$\frac{(x_3 - x_2)}{(x_2 - x_1)} = \frac{(y_3 - y_2)}{(y_2 - y_1)}$$

**Nota:**

En el problema planteado, para comparar dos números decimales considere que son iguales si la diferencia entre ellos es menor o igual a un margen de error establecido de 0,01.

**A continuación se presentan los siguientes casos de prueba:**

Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: 34  
El valor ingresado no cumple con lo solicitado

Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: 3  
El valor ingresado no cumple con lo solicitado

Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: 4  
Ingrese el punto A:2 6  
Ingrese el punto B:8 8  
Ingrese el punto C:4.77 6.94  
Ingrese el punto P:4 6  
Los puntos están alineados, no forman un triángulo

Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: 8  
Ingrese el punto A:8.26 8.6  
Ingrese el punto B:4.66 7.94  
Ingrese el punto C:3.05 7.64  
Ingrese el punto P:3.33 6.93  
Los puntos están alineados, no forman un triángulo  
Ingrese el punto A:4.33 -5.41  
Ingrese el punto B:-2.11 0.95  
Ingrese el punto C:-5.62 4.42  
Ingrese el punto P:-1.85 2.28  
Los puntos están alineados, no forman un triángulo

Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: 12  
Ingrese el punto A:-6.83 5.67



```

Ingrese el punto B:-4.32 -2.65
Ingrese el punto C:-1.96 -10.48
Ingrese el punto P:1 1
Los puntos están alineados, no forman un triángulo
Ingrese el punto A:-1.34 1.22
Ingrese el punto B:9.46 3.93
Ingrese el punto C:6.75 -3.68
Ingrese el punto P:6.09 1.48
El punto P(6.09,1.48) se encuentra dentro del triángulo formado por los puntos
A(-1.34,1.22) B(9.46,3.93) C(6.75,-3.68)
Ingrese el punto A:-9.47 -2.65
Ingrese el punto B:-4.75 3.18
Ingrese el punto C:-7.16 -9.3
Ingrese el punto P:-2 2
El punto P(-2.00,2.00) se encuentra fuera del triángulo formado por los puntos
A(-9.47,-2.65) B(-4.75,3.18) C(-7.16,-9.30)
La cantidad de puntos que se encontraron dentro de un triángulo son: 1
La cantidad de puntos que se encontraron fuera de un triángulo son: 1

```

### Programa 5: Propuesta de solución

```

1  #include <stdio.h>
2
3  double calcularOrientacion(double , double , double , double , double , double );
4  int alineados(double, double, double, double, double, double);
5
6  int main() {
7      int i=1,cantPuntos, contDentro=0, contFuera=0, estanAlineados;
8      double xi,yi,ax,ay,bx,by,cx,cy,px,py,orientacionABC,orientacionABP,orientacionBCP, orientacionCAP;
9
10     printf("Ingrese la cantidad de puntos, debe ser un múltiplo de 4 y encontrarse en el rango [4, 32]: ");
11     scanf("%d", &cantPuntos);
12
13     /*Validamos el ingreso de la cantidad de puntos*/
14     if (cantPuntos>=4 && cantPuntos<=32 && cantPuntos%4==0) {
15         while (i<=cantPuntos) {
16             printf("Ingrese el punto A:");
17             scanf("%lf %lf", &ax, &ay);
18             printf("Ingrese el punto B:");
19             scanf("%lf %lf", &bx, &by);
20             printf("Ingrese el punto C:");
21             scanf("%lf %lf", &cx, &cy);
22             printf("Ingrese el punto P:");
23             scanf("%lf %lf", &px, &py);
24             estanAlineados=alineados(ax,ay,bx,by,cx,cy);
25             if (!estanAlineados) {
26                 orientacionABC=calcularOrientacion(ax,ay,bx,by,cx,cy);
27                 orientacionABP=calcularOrientacion(ax,ay,bx,by,px,py);
28                 orientacionBCP=calcularOrientacion(bx,by,cx,cy,px,py);
29                 orientacionCAP=calcularOrientacion(cx,cy,ax,ay,px,py);
30                 /*Determinamos si el punto se encuentra dentro del triángulo, no se consideran los bordes*/
31                 if (orientacionABC>=0) {
32                     if (orientacionABP>=0 && orientacionBCP>=0 && orientacionCAP>=0) {
33                         printf("El punto P( %2.2lf, %2.2lf) se encuentra dentro del triángulo formado por
34                             los puntos A( %2.2lf, %2.2lf) B( %2.2lf, %2.2lf) C( %2.2lf, %2.2lf)\n",px,py,
35                             ax,ay,bx,by,cx,cy);
36                         contDentro++;
37                     } else {
38                         printf("El punto P( %2.2lf, %2.2lf) se encuentra fuera del triángulo formado por los
39                             puntos A( %2.2lf, %2.2lf) B( %2.2lf, %2.2lf) C( %2.2lf, %2.2lf)\n",px,py,ax,ay,
40                             bx,by,cx,cy);
41                         contFuera++;
42                     }
43                 } else if (orientacionABP<0 && orientacionBCP<0 && orientacionCAP<0) {
44                     printf("El punto P( %2.2lf, %2.2lf) se encuentra dentro del triángulo formado por los puntos
45                         A( %2.2lf, %2.2lf) B( %2.2lf, %2.2lf) C( %2.2lf, %2.2lf)\n",px,py,ax,ay,bx,by,cx,cy);
46                 }
47             }
48             i++;
49         }
50     }
51 }

```

```

41         contDentro++;
42     } else {
43         printf("El punto P( %2.2lf, %2.2lf) se encuentra fuera del triángulo formado por los puntos A
44             ( %2.2lf, %2.2lf) B( %2.2lf, %2.2lf) C( %2.2lf, %2.2lf)\n", px, py, ax, ay, bx, by, cx, cy);
45         contFuera++;
46     }
47     } else {
48         printf("Los puntos están alineados, no forman un triángulo\n");
49     }
50     i=i+4;
51 }
52 printf("La cantidad de puntos que se encontraron dentro de un triángulo son: %d\n", contDentro);
53 printf("La cantidad de puntos que se encontraron fuera de un triángulo son: %d\n", contFuera);
54
55 } else {
56     printf("El valor ingresado no cumple con lo solicitado\n");
57 }
58 return 0;
59 }
60
61
62 int alineados(double ax, double ay, double bx, double by, double cx, double cy) {
63     int alineados=0;
64     double vabx, vaby, vbcx, vbcy, div1, div2, diferencia;
65
66     /*AB=B-A*/
67     vabx=bx-ax;
68     vaby=by-ay;
69
70     /*BC=C-B*/
71     vbcx=cx-bx;
72     vbcy=cy-by;
73
74     div1= vbcx/vabx;
75     div2= vbcy/vaby;
76     diferencia=fabs(div2-div1);
77
78     if (diferencia<=0.01) {
79         alineados=1;
80     }
81     return alineados;
82 }
83
84 double calcularOrientacion(double ax, double ay, double bx, double by, double cx, double cy) {
85     return (ax-cx)*(by-cy)-(ay-cy)*(bx-cx);
86 }

```

### **Pregunta 3B (6 puntos) [propuesta por Jorge Berrocal]**

La resta es una operación aritmética que puede ser calculada con el método del complemento a nueve, método que se caracteriza en realizar dicha operación con tan solo la suma de dos números. Este método tiene la ventaja que la suma es una operación más sencilla de calcular que la resta, en especial para las computadoras, por lo que se convierte en una forma muy eficiente de realizar la operación.

Ahora bien, dados dos números enteros positivos, donde al número mayor lo denominaremos Minuendo mientras que al número menor lo denominaremos Sustraendo, como los mostrados en el siguiente ejemplo.

El método del complemento a nueve consiste en los siguientes pasos:

- Tomar el Sustraendo de la operación y cambiar cada dígito del Sustraendo por el resultado del complemento a nueve del dígito. Para el ejemplo 5,932 deberá ser reemplazado por 4,067. Se puede observar que cada dígito del Sustraendo es reemplazado por la diferencia entre 9 y dicho dígito.

$$\begin{array}{r}
 4,687,945 - \\
 5,932 \\
 \hline
 4,682,013
 \end{array}$$

Figura 8: Ejemplo de una Resta mostrando Minuendo, Sustraendo y el resultado.

- Como el método requiere que tanto el Minuendo como el Sustraendo tengan la misma cantidad de dígitos se debe completar al número obtenido en el paso anterior con nueves por la izquierda hasta completar la cantidad de dígitos del Minuendo. Para el ejemplo 4,067 se convertirá en 9,994,067, dado que el Minuendo del ejemplo (4,687,945) tiene siete dígitos. De esta manera hemos conseguido el complemento a nueve del Sustraendo.
- El siguiente paso consiste en sumar al Minuendo el complemento a nueve del Sustraendo obtenido en el paso anterior. En el ejemplo al sumar 4,687,945 y 9,994,067 se obtiene 14,682,012.
- El método termina omitiendo el primer dígito de la izquierda (que siempre será 1) y sumándole uno a la cantidad obtenida. Por lo que en el ejemplo al número 14,682,012 primero se le omite el primer dígito de la izquierda obteniendo 4,682,012 para después sumarle uno al número obteniendo 4,682,013 número que es el resultado de la resta del Minuendo y Sustraendo dados como ejemplo.

**Se pide desarrollar un programa en lenguaje C** que permita leer el Minuendo y Sustraendo de una resta. Deberá validar que ambos números son enteros mayores iguales a cero. En caso alguno de dichos números no lo fuera deberá entregar el mensaje “El Minuendo y Sustraendo deben ser enteros mayores iguales a cero.” Además, deberá validar que el Minuendo es mayor o igual al Sustraendo. Si no lo fuera deberá entregar el mensaje “El Minuendo debe ser mayor o igual al Sustraendo.”

El programa deberá hacer uso de dos módulos además del módulo principal. El primer módulo deberá entregar dado un número como parámetro la cantidad de dígitos que tiene dicho número, mientras que el segundo módulo deberá calcular y entregar el complemento a nueve del Sustraendo, recibiendo también como parámetros la cantidad de dígitos del Minuendo y del Sustraendo.

Por último, el módulo principal deberá calcular el resultado de la resta siguiendo el método del complemento a nueve, e imprimir los resultados obtenidos.

#### Tomar en cuenta:

- No deberá implementar otros módulos diferentes a los indicados previamente.

#### Recordar que:

El número de dígitos de un número X es igual a la parte entera del  $\log_{10}(X) + 1$ .

#### Casos de prueba para verificar la solución:

Ingrese el Minuendo y Sustraendo de la operación: 453444224 -5467  
El Minuendo y Sustraendo deben ser enteros mayores iguales a cero.

Ingrese el Minuendo y Sustraendo de la operación: -675354 -56765  
El Minuendo y Sustraendo deben ser enteros mayores iguales a cero.

Ingrese el Minuendo y Sustraendo de la operación: 4835 934735  
El Minuendo debe ser mayor o igual al Sustraendo.

Ingrese el Minuendo y Sustraendo de la operación: 4687945 5932

El complemento del Sustraendo es 9994067.  
La suma del Minuendo y el complemento a nueve del Sustraendo es 14682012.  
La resta del Minuendo menos el Sustraendo usando el complemento a nueve es 4682013.

Ingrese el Minuendo y Sustraendo de la operación: 942053495 0

El complemento del Sustraendo es 999999999.  
La suma del Minuendo y el complemento a nueve del Sustraendo es 1942053494.  
La resta del Minuendo menos el Sustraendo usando el complemento a nueve es 942053495.

### Programa 6: Propuesta de solución

```
1  #include <stdio.h>
2  #include <math.h>
3
4  //Declaración de subprogramas.
5  int ObtenerCantidadDigitos(int);
6  int ObtenerComplemento(int, int, int);
7
8  int main ()
9  {
10     int Minuendo, Sustraendo, CantDigitosMinuendo, CantDigitosSustraendo, ComplementoSustraendo, Suma, Resta;
11
12     //Ingresa el Minuendo y Sustraendo de la operación.
13     printf("Ingrese el Minuendo y Sustraendo de la operacion: ");
14     scanf("%d %d", &Minuendo, &Sustraendo);
15
16     //Verifica que ambos número sean positivos.
17     if ( Minuendo >= 0 && Sustraendo >= 0 )
18     {
19         //Verifica que el Minuendo es mayor que el Sustraendo.
20         if ( Minuendo >= Sustraendo )
21         {
22             //Obtiene la cantidad de dígitos del Minuendo y el Sustraendo.
23             CantDigitosMinuendo = ObtenerCantidadDigitos(Minuendo);
24             CantDigitosSustraendo = ObtenerCantidadDigitos(Sustraendo);
25
26             //Obtiene el complemento a nueve del sustraendo, con la cantidad de dígitos del Minuendo.
27             ComplementoSustraendo = ObtenerComplemento(Sustraendo, CantDigitosMinuendo, CantDigitosSustraendo);
28             ;
29
30             //Obtiene la suma del Minuendo más el complemento a nueve del Sustraendo así como la resta.
31             Suma = Minuendo + ComplementoSustraendo;
32             Resta = Suma - pow(10, CantDigitosMinuendo) + 1;
33
34             //Imprime los resultados del problema.
35             printf("\nEl complemento del Sustraendo es %d.\n", ComplementoSustraendo);
36             printf("La suma del Minuendo y el complemento a nueve del Sustraendo es %d.\n", Suma);
37             printf("La resta del Minuendo menos el Sustraendo usando el complemento a nueve es %d.", Resta);
38         }
39         else
40             printf("El Minuendo debe ser mayor o igual al Sustraendo.");
41     }
42     else
43         printf("El Minuendo y Sustraendo deben ser enteros mayores iguales a cero.");
44
45     return 0;
```

```

45 }
46
47 //Devuelve el número de dígitos que tiene el parámetro Numero.
48 int ObtenerCantidadDigitos(int Numero)
49 {
50     int Cont;
51
52     //Utilizando la función logaritmo para calcular el número de dígitos.
53     Cont = (int)(log10(Numero) + 1);
54
55     return Cont;
56 }
57
58 //Función que entrega el complemento a nueve de un número.
59 int ObtenerComplemento(int Numero, int CantMinuendo, int CantSustraendo)
60 {
61     int Complemento=0, Cont=0, Dígito;
62
63     /* Alternativa de solución con una iteración y condicional. */
64     while ( Cont < CantMinuendo )
65     {
66         //Si Cont es menor que CantSustraendo.
67         if ( Cont < CantSustraendo )
68         {
69             //Obtiene el dígito de menor valor de Numero.
70             Dígito = Numero % 10;
71             Numero = Numero / 10;
72
73             //Obtiene el complemento a 9 de Dígito y lo acumula en Complemento.
74             Dígito = 9 - Dígito;
75             Complemento = Complemento + Dígito*pow(10, Cont);
76         }
77         //Si Cont es mayor igual a CantSustraendo.
78         else
79             //Agrega un 9 a la derecha a Complemento.
80             Complemento = Complemento + 9*pow(10, Cont);
81
82         //Incrementa en uno a Cont.
83         Cont++;
84     }
85
86     /* Alternativa de solución con dos iteraciones.
87     while ( Cont < CantSustraendo )
88     {
89         //Obtiene el dígito de menor valor de Numero.
90         Dígito = Numero % 10;
91         Numero = Numero / 10;
92
93         //Obtiene el complemento a 9 de Dígito y lo acumula en Complemento.
94         Dígito = 9 - Dígito;
95         Complemento = Complemento + Dígito*pow(10, Cont);
96
97         //Incrementa en uno a Cont.
98         Cont++;
99     }
100
101     while ( Cont < CantMinuendo )
102     {
103         //Agrega un 9 a la derecha a Complemento.
104         Complemento = Complemento + 9*pow(10, Cont);
105
106         //Incrementa en uno a Cont.
107         Cont++;
108     } */
109
110     return Complemento;
111 }

```

---

El archivo debe ser grabado con el nombre **E1\_código\_pregunta3.c** (donde la palabra código debe ser reemplazada por el código del alumno, por ejemplo, **E1\_20206666\_pregunta3.c**, sin espacios).

San Miguel, octubre del 2021