

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

4ta. práctica (tipo b)
(Primer Semestre 2017)

Indicaciones Generales:

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase.
- No se pueden emplear **variables globales, estructuras, ni objetos** (con excepción de "cin" y "cout"). Tampoco se podrán emplear las funciones malloc, realloc, strdup, strtok, sscanf, sprintf ni fopen.
- Deberá modular correctamente el proyecto en archivos independientes. El archivo **main.cpp** solo podrá contener la función **main** de cada proyecto.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- La **presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Cuestionario:

Una institución tiene en un archivo de textos todos los registros de las llamadas telefónicas que se realizan desde los anexos colocados en las diferentes oficinas de sus trabajadores durante un tiempo determinado. El archivo es similar al que se muestra a continuación:

20780023	8153	10.8543	58643376	9735	8.85	27709566	2535	15.53	20780023	1234
60.10	15290194	9958	120.85	...						
...										

En el archivo se registra el **código de la unidad** a la que pertenece el anexo, el **número de anexo**, y el **tiempo de la llamada** en segundos. Como se puede apreciar en el archivo, los datos vienen agrupados de tres en tres con esa información. Como se trata de registros históricos de las llamadas realizadas, la **unidad puede aparecer muchas veces en el archivo, de igual manera que el anexo**. Por ello, se desea que usted confeccione una aplicación en C/C++ en la que se pueda administrar estos datos de manera más ordenada. El ingreso de datos se realizará por re-direccionamiento de la entrada estándar empleando únicamente la biblioteca "iostream" de C++.

Pregunta 1 (12 puntos)

Leer los datos del archivo y guardarlos en **arreglos unidimensionales de memoria dinámica**, de modo que se forme una "tabla indexada" (figurativamente, usando el conjunto de los arreglos). La figura siguiente muestra lo que se pretende:

	Unidad	Índice inicial	Índice final
1	20780023	0	23
2	58643376	24	24
3	27709566	25	76
4	15290194	77	103
...

	Anexo	Tiempo	
1	8153	10.8543	} Anexos de la unidad 20780023
2	1234	60.10	
...	
22	1234	507.91	} Anexo de la unidad 58643376
23	5791	125.84	
24	9735	8.85	
25	2535	15.53	} Anexos de la unidad 27709566
...	
76	8899	273.94	
77	9958	120.85	
...	

Es importante resaltar que no se conoce la cantidad total de unidades, anexos, ni llamadas telefónicas, por lo que el espacio de almacenamiento deberá administrarse por **incrementos constantes**.

Respecto al proceso de creación de las tablas, este debe hacerse de modo tal que cuando se lea un juego de datos se **busque el valor de la unidad leída** en el arreglo denominado "**unidad**". Si esta no se encuentra, debe colocar la nueva unidad **al final de los datos del arreglo**.

Luego, debe proceder de manera similar para los otros datos (anexo y tiempo), pero teniendo en consideración los valores de los arreglos de los índices (inicial, final) que también deben ir actualizándose durante la lectura. Estos valores representan el inicio y final de las posiciones donde se guardan los valores de anexo y tiempo de cada unidad. De este modo, en caso de que la unidad ya está en el arreglo, el anexo y tiempo deberán colocarse al final de la "sub-lista" de llamadas de esa unidad, **desplazando los valores que se encuentren por debajo de ella, y corrigiendo los índices** de todas las unidades involucradas.

Asimismo, deben tener en cuenta que en esta etapa no se realizarán acciones especiales con respecto a los anexos repetidos, sólo se colocarán al final de sus respectivas listas como cualquier otro. Por otro lado, después de haber construido los arreglos con las dimensiones exactas, el programa deberá imprimir los datos de la "tabla indexada" de una manera similar a la siguiente:

1) Unidad: 20780023	(Debe imprimir la numeración)
Índice inicial: 0	
Índice final: 23	
Llamadas:	
Anexo	Tiempo
1) 8153	10.854
2) 1234	60.10
...	...
2) Unidad: 58643376	
Índice inicial: 24	
...	

Nota: no se podrán emplear más arreglos de los que aparecen en la figura, salvo para el momento en que se requiera redimensionar los arreglos.

Pregunta 2 (8 puntos)

Prepare un nuevo proyecto, copie el código hecho en la pregunta anterior, y agregue el necesario para solucionar el siguiente enunciado:

Una vez realizada la pregunta anterior, se deberá resumir las tablas de llamadas y actualizar los índices. La idea es que en la lista de llamadas **no se encuentren anexos repetidos** en las unidades, y que los **tiempos se acumulen**. Para realizar esta acción, deberá trabajar con los arreglos "anexo" y "tiempo".

Para cada unidad, deberá tomar cada uno de sus anexos que le correspondan, de acuerdo al arreglo de índices, y a partir de allí buscar los duplicados en su "sub-lista". Si se encuentran anexos repetidos, se deberá acumular el tiempo y eliminar el registro duplicado. En estos casos, **se moverán todos los datos hacia arriba (actualizándose los índices)**. Asimismo, si el espacio libre final en los arreglos es igual al incremento empleado en la pregunta anterior, se deberá reducir el espacio físico de los arreglos.

La figura siguiente muestra el proceso solicitado:

Estado inicial			Luego de encontrar el primer repetido		
	Anexo	Tiempo		Anexo	Tiempo
1	8153	10.8543	1	8153	10.8543
2	1234	60.10	2	1234	568.01
...
22	1234	507.91	22	5791	125.84
23	5791	125.84	23	9735	8.85
24	9735	8.85	24	2535	15.53
25	2535	15.53	25
...	8899	273.94
76	8899	273.94	76	9958	120.85
77	9958	120.85	77
...			

Finalmente, se deberá emitir un reporte similar al de la pregunta anterior, pero con los acumulados calculados al final de cada unidad.

Nota: no se podrán emplear más arreglos de los que aparecen en la figura, salvo para el momento de redimensionar los arreglos.

Consideraciones finales:

- Cada pregunta debe ser desarrollada en un proyecto independiente, debe denominar a cada proyecto como "Pregunta1" y "Pregunta2". Si el alumno presenta solo un proyecto, la calificación de la prueba no superará el puntaje de la primera pregunta.
- Cree en el computador una carpeta de trabajo con la siguiente ruta: c:\temp\Laboratorio4. En ella colocará los proyectos que den solución a los problemas planteados.
- **En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código. De no hacerlo se le descontará 0.5 puntos por archivo.**

Al finalizar la práctica, comprima¹ la carpeta ***laboratorio4*** en un archivo con nombre <código del alumno con 8 dígitos>.<extensión del archivo comprimido> y súbalo a la Intranet del curso, en el enlace Documentos, en la carpeta \Laboratorio4\<código del horario>\<aula>.

Profesores del curso: Arturo Oncevay
 Miguel Guanira

San Miguel, 5 de mayo del 2017.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip).