

Licenciatura en Ingeniería de Software
“Programación Estructurada”
M.G.T.I. María Enriqueta Castellanos Bolaños
Tercer Parcial

Fecha: 29 de junio de 2021

Nombre(s): _____ Calificación: _____

Nombre del archivo enviado por correo electrónico: _____

Instrucciones generales:

- Lee cuidadosamente todas las instrucciones antes de iniciar.
- La práctica tiene un valor de 30 puntos y representa el 30% de la calificación final del curso.
- Dispones de 1 hora con 15 minutos para resolver la práctica.
- Durante la práctica:
 - No se resolverán dudas relacionadas sobre cómo codificar el programa.
 - No podrás solicitar que se evalúe el procedimiento que hasta el momento hayas desarrollado.
 - Deberás resolver por ti mismo los errores de compilación que se te presenten.
 - No está permitido mantener comunicación con otros estudiantes.
 - Está prohibido consultar fuentes impresas y/o electrónicas.
- Queda estrictamente prohibido el uso de teléfonos celulares.
- Asegúrate de nombrar el archivo de código fuente de acuerdo al siguiente formato: <nombre del alumno en minúsculas y sin espacios en blanco><apellido paterno en minúsculas y sin espacios en blanco>. Por ejemplo, para alguien cuyo nombre sea Juan Pérez Ruiz, el nombre de su archivo sería: juanperez.cpp.
- El código fuente deberá tener una indentación o sangría de 3 espacios en blanco.
- Una vez que hayas finalizado el examen, deberás depositar el archivo de código fuente (.cpp) en la sección **Exámenes parciales** del curso en línea.
- Únicamente serán revisados aquellos programas que compilen. Se recomienda comentar las líneas de código que arrojen mensajes de error, con el fin de no perder la totalidad de los puntos.
- No se admitirán correcciones al programa una vez que hayas hecho entrega formal de éste en la plataforma.

I. Instrucciones: Analiza la situación que a continuación se plantea y desarrolla un programa que se ajuste a lo especificado en los incisos a, b, c, d y e:

Enrique desea implementar un programa en C que reciba como entrada una frase de texto (con una longitud máxima de 25 caracteres, incluyendo el carácter nulo), convierta las vocales presentes en mayúsculas y efectúe un conteo del número total de ellas. La frase deberá imprimirse en pantalla carácter por carácter junto con el número total de vocales halladas. Es importante mencionar que el programa tendrá la capacidad de reconocer entre vocales mayúsculas y minúsculas. Utiliza las funciones: `int islower(int argument)`, `int toupper(int argument)` incluidas en la librería `ctype.h`.

Ejemplo

Si un usuario introduce la frase: *hoy es tu día de suerte*

Ésta quedaría transformada en: *hOy Es tU dÍA dE sUErtE*

Y el número total de vocales encontradas sería: **9**

Considera el fragmento de código fuente que a continuación se presenta como parte de la solución del problema, y añade las sentencias necesarias que permitan efectuar el proceso solicitado.

```
#include <stdio.h>
#include <string.h>
...

bool validar(char *ptrFrase);
int convertir(char *ptrFrase);
void imprimir(char *ptrFrase);

int main(){

    char frase[MAX_CARACTERES];
    ...

    return 0;
}
```

Las tareas que el programa deberá realizar se describen con detalle en los siguientes puntos:

- a) Verificar que el tamaño de la frase no exceda los 25 caracteres, incluyendo el carácter nulo (`#define MAX_CARACTERES 25`). Para ello deberás crear una función que se ajuste al prototipo que se expone seguidamente e invocarla mediante una estructura **do while**, desde la función principal del programa (**8 puntos**).

```
bool validar(char *ptrFrase)
```

- b) Convertir las vocales minúsculas de la frase en mayúsculas y realizar un conteo del número de ellas. Para ello deberás crear una función que se ajuste al prototipo que se expone seguidamente. Es indispensable que la función utilice la notación apuntador/desplazamiento (**12 puntos**):

```
int convertir(char *ptrFrase)
```

- c) Imprimir la frase modificada carácter por carácter utilizando una sentencia de iteración. Para ello deberás crear una función que se ajuste al prototipo que se expone seguidamente. Es indispensable que la función utilice la notación apuntador/desplazamiento (**9 puntos**):

```
void imprimir(char *ptrFrase)
```

- d) Imprimir el número total de vocales encontradas en la frase.
- e) Indentar adecuadamente el código fuente (**1 punto**). La puntuación será asignada sólo si el código fuente compila.

Consideraciones importantes:

- La indentación del código fuente deberá ser de 3 espacios en blanco.
- El programa deberá ser implementado de acuerdo a lo establecido en los incisos anteriores. Si no cumple esos criterios, será imposible verificar el dominio de los conceptos evaluados.
- No serán otorgadas proporciones de la puntuación asignada a cada inciso, por implementaciones incorrectas, incompletas o mal estructuradas.
- Se descontarán puntos por incurrir en malas prácticas de programación.

Criterios de evaluación:

- Cumplimiento de los lineamientos establecidos en la prueba.
- Planteamiento claro y coherente de la solución a la problemática presentada.
- Dominio en la aplicación de los conceptos fundamentales de la metodología de la programación en la solución de problemas.
- Implementación correcta de la solución al problema planteado.
- Uso correcto de las instrucciones del lenguaje de programación de alto nivel.
- Uso de una herramienta de depuración de código fuente.
- Ausencia de errores sintácticos (o de compilación).
- Ausencia de errores semánticos.
- Desempeño exitoso de la ejecución del programa.
- ~~Uso de estándares de codificación.~~
- Seguimiento de buenas prácticas de programación.