

Tarea 1. Algoritmos y Estructuras de Datos I

Fecha de Publicación: 25/Feb/2016 - **Fecha límite de Entrega 24:00 hrs. del 6/Mar/2016**

Indicaciones:

- La tarea se realizará por equipos de 1 a 3 personas, la calificación será más estricta dependiendo del número de integrantes en el equipo.
- Todos los ejercicios se deberán de ser implementados.
- Como entregables se entregará un archivo por ejercicio para los ejercicios 1 - 4 y tres o mas archivos para el ejercicio 5, éstos archivos deberán contener el código fuente de la implementación. Además se deberá entregar un documento PDF en el cual se describa brevemente la solución dada a cada uno de los problemas.
- Los archivos se empaquetarán en un archivo zip y se enviarán por correo. El asunto del correo será [Tarea 01] (ojo poner los corchetes). En el contenido del correo se deberán escribir los nombres de los integrantes del equipo.
- **Por ningún motivo se recibirán tareas de forma extemporánea a la fecha límite.** Evite esperar hasta el último momento para enviar la tarea y así evitar contratiempos.

Ejercicios

1.- Escriba una función de nombre `printArray` que reciba un apuntador a un arreglo de enteros y regrese nada. La función debe de imprimir en pantalla el contenido del arreglo con la cadena `" , "` entre cada par de elementos, pero no así después del último elemento del arreglo. (1 Punto)

2.- Escriba una función de nombre `reverse` que reciba un apuntador a un arreglo de enteros y regrese nada. La función deberá colocar los elementos del arreglo de manera inversa dentro del mismo arreglo. (2 Puntos)

3.- Implementar un programa que imprima los primeros N números de la sucesión de Fibonacci. La sucesión de Fibonacci se define como una sucesión infinita de números naturales que comienza con los enteros 0, 1. Los siguientes términos se definen como la suma de los dos anteriores. Esto es, $f(n) = f(n-1) + f(n-2)$. Por ejemplo:

Para $n=4 \Rightarrow 0, 1, 1, 2$.

Para $n=11 \Rightarrow 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55$

El programa deberá contener la siguiente estructura básica:

```
main() {  
    pedir un número  
    N <- leer numero ingresado  
    A <- Fibonacci (N)    // donde A es un apuntador a un arreglo  
    imprime A  
}  
  
A* Fibonacci (N) {  
    A <- calculo de la sucesión  
    regresa A  
}
```

Hint: no hacerlo de forma recursiva, sea lo que sea que signifique recursivo (2 Puntos)

Tarea 1. Algoritmos y Estructuras de Datos I

4.- Implementar una función de nombre `strlen` que reciba una cadena y regrese un entero. La función deberá de calcular el tamaño de la cadena dada. Queda prohibido hacer uso de funciones ya implementadas para este fin como la función `strlen` implementada en la librería `string.h`, además no se permite acceder a los elementos del arreglo con el operador `[]`. Hint: revisar como se define en C el fin de una cadena expresada como un arreglo de caracteres. (2 Puntos)

5.- Implementar un tipo de dato abstracto `Polygon` el cual representará la forma geométrica polígono. La estructura deberá de implementar 3 funciones básicas de los polígonos: 1) encontrar el perímetro del polígono, 2) encontrar el área del polígono y 3) obtener el centroide del polígono. La definición de dichas funciones será la siguiente:

```
float perimeter(Polygon *p);  
float area(Polygon *p);  
Point * centroid(Polygon *p);
```

Además se deberá de implementar una función `main` donde se cree una instancia de la estructura `Polygon` y se prueben las tres funciones imprimiendo en pantalla el resultado. La implementación se deberá hacer en tres archivos distintos, uno para la interfaz, otro para la implementación y uno para la función `main`.

Hint: la definición de polígono (<https://es.wikipedia.org/wiki/Pol%C3%ADgono>). El área de un polígono se puede calcular con el método de Gauss (https://es.wikipedia.org/wiki/F%C3%B3rmula_del_%C3%A1rea_de_Gauss).

Revisar lo visto en clase para la estructura `Point` necesaria para implementar el perímetro y el centroide

(3.5 Puntos).