



Tercera Práctica Calificada: 12/11/2018

CC112-CD

Ciclo: 2018-2

Normas:

1. El alumno entregará esta hoja de examen debidamente llenada con sus datos.
2. La solución de la prueba se guardarán en **Escritorio**, carpeta: **ApellidoNombreCodigo** (sin espacios en blanco), la pregunta **n** se guardará en el archivo: **n.c** ($n = 1, 2, \dots$).
3. No se permite: El uso de celulares, internet, USB, ingresar después de 15 min. de iniciado el examen ni salir antes de la hora de finalización.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos : _____ Nombres : _____
Sección : _____ Grupo: _____

1. [5 ptos.] Escriba una función que reciba una referencia a un arreglo de enteros con al menos un elemento igual a cero, muestre todos los elementos de dicho arreglo hasta antes del primer elemento igual a cero. También escriba otra función tal que al pasarle un par de referencias a arreglos de enteros con al menos un elemento igual a cero, copie todos los elementos hasta antes del primer elemento igual a cero del arreglo de la segunda referencia en los elementos del arreglo de la primera referencia. Para ello debe completar en los puntos suspensivos:

```
#include <stdio.h>
void copiar(int * des, int * ori);
void mostrar(int * p);
int main(){
    int destino[5] = {4,5,6,7,0};
    int origen[4] = {-1,-3,-2,0};
    mostrar(destino);
    copiar(destino, origen);
    mostrar(destino);
}
void mostrar(int * p){/*...*/;}
void copiar(int * des, int * ori){/*...*/;}
```

Para el ejemplo, La salida debe ser:

```
4 5 6 7
-1 -3 -2 7
```

2. [5 ptos.] Escriba una función tal que al pasarle una referencia a un arreglo de enteros formado enteros diferentes de cero, salvo el último término, muestre todos los elementos distintos de cero de dicho arreglo. También escriba otra función tal que al pasarle una referencia a un arreglo de enteros formado enteros diferentes de cero, salvo el último término, retorne la cantidad de elementos distintos de cero de dicho arreglo. Luego, implemente una función tal que al pasarle un par de referencias a arreglos de enteros diferentes de cero, salvo el último término, anexe, desde la posición del elemento igual a cero del arreglo referenciado por la primera referencia, todos los elementos del arreglo referenciado por la segunda referencia. Para ello debe completar en los puntos suspensivos:

```
#include <stdio.h>
void anexar(int * des, int * ori);
void mostrar(int * p);
```

```

int longitud(int * p);
int main(){
    int destino[6] = {4,5,0};
    int origen[4] = {-1,-3,-2,0};
    mostrar(destino);
    anexar(destino, origen);
    mostrar(destino);
}
int longitud(int * p){/*...*/;}
void mostrar(int * p){/*...*/;}
void anexar(int * des, int * ori){/*...*/;}

```

Para el ejemplo, La salida debe ser:

```

4 5
4 5 -1 -3 -2

```

3. [5 ptos.] En un salón de clase hay un letrero de 4 líneas:

La hora es la hora
 Ante de la hora no es la hora
 Después de la hora tampoco es la hora
 La hora es la hora.

Escriba un programa que cuente las letras mayúsculas y las minúsculas, así como el número de blancos y puntos y finalmente el número total de caracteres

Sugerencia guarde el letrero en **char letrero[4][40]**.

La salida será:

Número de mayúsculas: 4
 Número de minúsculas: 75
 Número de blancos: 22
 Número de puntos: 1
 Número total de caracteres: 102

4. [5. ptos.] Escriba un programa que valide (verifique) el formato de una clave de correo electrónico que contenga caracteres MMmmm*nnn:

elementos en mayúsculas: 1<= MM <= 2

elementos en minúsculas: 3<=mmm

Número de '*': nAst == 1

elementos numéricos >= 4

No deben haber otros elementos, como blanco, '', '?', etc.

Los elementos de MM, mmm y nn pueden estar entremezclados.

Por ejemplo para: [Fiore19*83](#), la salida debe ser:

Clave correcta: [Fiore19*83](#)