

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN
EJERCICIOS DE APUNTADES

1. Escriba un programa que capture dos números enteros. Después debe mandar una función de nombre `swap(x, y)` que recibe dos enteros. La función debe regresar el valor más grande de los números, e intercambiar los valores que contienen las variables que se le pasaron como parámetros. Se deben imprimir los valores antes y después de haber mandado llamar la función, así como el valor del número más grande.
2. Escriba un programa que capture todos los elementos de un arreglo de tipo entero sin signo. La longitud del arreglo es de 20 elementos. Una vez capturados el programa debe desplegar un menú con las siguientes opciones: Búsqueda, Modificación, Listado, Fin. Cada opción debe mandar llamar una función que recibe como parámetro el arreglo capturado. La función de búsqueda debe preguntar por un número y buscarlo dentro del arreglo; si lo encuentra debe regresar el lugar donde se encuentra, en caso contrario debe regresar un -1. Se debe desplegar esta información o un mensaje indicando que el número no existe. La función de modificación va a preguntar por un número de ubicación (0 a 19), desplegará el contenido de esta ubicación y preguntará por un nuevo valor para substituir el que ahí se encontraba. La función de listado desplegará el contenido del arreglo. La opción de Fin termina el programa.
3. Escriba un programa que lea un arreglo de 10 números flotantes. El programa debe contar con tres funciones. Una función de nombre `mayor()` debe regresar el número más grande del arreglo. Otra función de nombre `menor()` debe regresar el valor más pequeño del arreglo. La tercera función de nombre `promedio` debe regresar el promedio de los 10 números. Se deben mandar llamar dichas funciones y desplegar su resultado.
4. Escribir una función que convierta una cadena `s` a un número de punto flotante usando apuntadores. Considerar que el número tiene el siguiente formato `99999999.999999`, es decir, no se dará en notación científica. La función deberá suministrársele una cadena y deberá devolver un número.
5. Escribir un programa que encuentre el número de veces que una palabra dada (esto es, una cadena corta) ocurre en una sentencia (una cadena larga).

Leer los datos de la entrada estándar. La primera línea es una sola palabra, en la segunda línea se tiene un texto general. Leer ambas hasta encontrar un carácter de nueva línea. Recordar que se debe insertar un carácter nulo antes de procesar.

La salida típica podría ser:

La palabra es "el"

La sentencia es "el perro, el gato y el canario"
La palabra ocurrio 3 veces.

6. Dado el siguiente programa, ejecute el mismo a mano e indique los resultados que genera.

```
void p1(int *i,int j,int k);  
void p2(int h,int *j,int k);  
void p3(int *i);
```

```
main()  
{  
  int i,j,k;  
  i =0;  
  j=1;  
  k=2;  
  p2(0,&k);  
  p2(1,&i);  
  p2(2,&j);  
}
```

```
void p1(int *i,int j,int k)  
{  
  (*i)++;  
  printf(“%d %d %d”,*i,j,k);  
}
```

```
void p2(int h,int *j,int k)  
{  
  int i;  
  i=j;  
  if (h==0)  
    p1(*j,j,k)  
  else  
    if (h==1)  
      p1(&i,*j,k);  
  else  
    p3(&i,*j,k);  
  printf(“%d %d %d”,i,j,k);  
}
```

```
void p3(int *i)  
{  
  (*i)++;  
}
```

7. a) Declare una estructura llamada **condominio** con los siguientes componentes:

cantidad_vecinos: de tipo entero,

avenida: de tipo arreglo de N caracteres,

calle: de tipo arreglo de P caracteres,

nombre_edificio: de tipo arreglo de M caracteres,

gasto_mensual: de tipo real,

superficie_abitable: de tipo real

b) Declare las variables **mi_edificio** y **tu_edificio** del tipo estructura **condominio**.

c) Escriba las instrucciones necesarias para realizar las siguientes acciones:

1 Asignar el valor 3 a **cantidad_vecinos** de la variable **mi_edificio**

2 Asignar el valor 'C' al componente **avenida** de la variable **tu_edificio**

3 Asignar el valor 2000,50 al componente **superficie_abitable** de la variable **tu_edificio**.

d) Escribir el valor del componente **gasto_mensual** de la variable **mi_edificio**