



Tecnicatura Superior en Análisis, Desarrollo y Programación **de Aplicaciones** **Programación I**

“Ejercicios adicionales sobre punteros y funciones”
Versión 1.0 Julio 2016

Lic. Guillermo R. Cherencio



Implemente y utilice (realice un “testing” desde un programa principal o función main()) las siguientes funciones:

Función		Objetivo	Prototipo
swap		Intercambiar el contenido de los argumentos que recibe. El valor apuntado por el segundo argumento pasa al primero y el primero al segundo.	void swap(int *,int *)
	Ejemplo de uso	<pre>int a=4,b=5; swap(&a,&b); printf(“%d %d\n”,a,b); int *pa = &a; int *pb = &b; swap(pa,pb); printf(“%d %d\n”,a,b);</pre>	
	Salida esperada	<pre>5 4 4 5</pre>	
swapall		Idem swap() pero esta función permite intercambiar cualquier tipo de dato.	void swapall(void *,void *,int size)
	Ejemplo de uso	<pre>int a=4,b=5; swapall(&a,&b,sizeof(int)); double d1=4.25,d2=3.15; swapall(&d1,&d2,sizeof(double)); printf(“%d %d\n”,a,b); printf(“%lf %lf\n”,d1,d2);</pre>	
	Salida esperada	<pre>5 4 3,150000 4,250000</pre>	
reverse		Pone a la cadena en orden inverso. En el ejemplo se utilizó strdup() porque reverse() no puede realizarse sobre constantes ¹ .	void reverse(char *)
	Ejemplo de uso	<pre>char *mensaje = strdup(“hola”); reverse(mensaje); printf(“%s\n”,mensaje); free(mensaje);</pre>	

¹ Una constante de cadena (por ejemplo: char *m = “mensaje”;) no puede ser modificada, solo puede ser leída; se debe utilizar malloc o strdup (quien llama a malloc) para crear una cadena “modificable” en memoria (por ejemplo: char *m = strdup(“mensaje”);); no obstante, no debe olvidarse luego de ser usada, hacer el free() correspondiente.



	Salida esperada	aloh
--	------------------------	------

Función		Objetivo	Prototipo
strsearch		Busca carácter dentro de cadena de caracteres y devuelve la posición en la que se encuentra el carácter. Si el carácter no se encuentra, devuelve -1.	int strsearch(char *,char)
	Ejemplo de uso	char *mensaje = "hola maria"; int pos = strsearch(mensaje,'a'); int pos2 = strsearch(mensaje,'w') printf("%d %d\n",pos,pos2);	
	Salida esperada	3 -1	
strpsearch		Idem strsearch(), pero en vez de devolver la posición del carácter, devuelve un puntero a la posición en donde se encuentra el carácter. Si el carácter no se encuentra, devuelve NULL.	char * strpsearch(char *,char)
	Ejemplo de uso	char *mensaje = "hola maria"; char *pos = strsearch(mensaje,'a'); char *pos2 = strsearch(mensaje,'w') printf("[%s] %p\n",pos,pos2);	
	Salida esperada	[a maria] NULL	

Atte. Guillermo Cherencio
 Programación I
 ISFT N° 189