

Pregunta 1

Parte a.- Programe la función:

```
char *traducir(char *s, char *t, char *r);
```

Esta función debe entregar un nuevo string que se obtiene reemplazando en *s* todos los caracteres que aparezcan en *t* por el caracter que aparezca en la misma posición en el string *r*. El largo de *t* es siempre igual al largo de *r*. Ud. no puede usar el operador de indexación [].

Ejemplo:

```
char *t= traducir("HOLA QUE TAL", "AEIOU", "12345");
/* t es "H4L1 Q52 T1L" */
```

Parte b.- Programe la función:

```
char *ultimaDireccionValida(char *ptr);
```

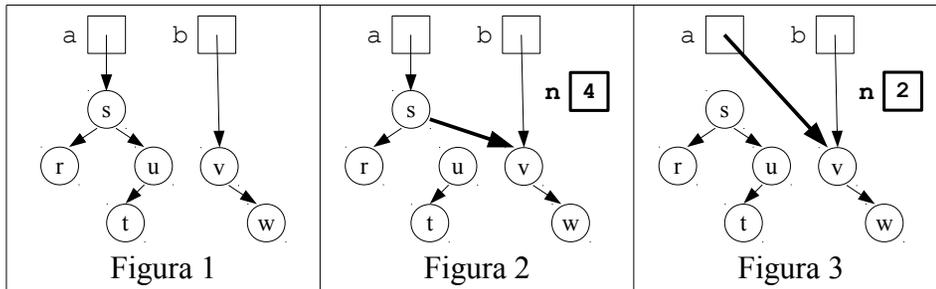
Esta función debe entregar la última dirección válida que se puede leer a partir de *ptr*. Para calcularla lea el caracter apuntado por *ptr* e incremente *ptr* en 1. Repita esta lectura e incremento indefinidamente hasta que se produzca el *segmentation fault*. Capture la señal *SIGSEGV*. Resguarde *ptr* en una variable global. El último valor de *ptr* menos 1 es la última dirección válida.

Pregunta 2

Programe la función:

```
int reemplazarNodoK(Nodo **pa, int k, Nodo *b);
```

Sea $a = *pa$. Esta función reemplaza el *k*-ésimo nodo del árbol *a* por el nodo *b*. El *k*-ésimo nodo de *a* es el *k*-ésimo nodo al recorrer los nodos de *a* en orden¹. Por ejemplo en la figura 1, *r* es el primer nodo, *s* el segundo, *t* el tercero, *u* el cuarto. Esta función retorna *k* si se hizo el reemplazo, es decir cuando el árbol *a* tenía al menos *k* nodos. Si no, entrega el número de nodos encontrados en *a*, que será inferior a *k*. Las siguientes figuras sirven para explicar algunos ejemplos de uso. Los punteros *a* y *b* son de tipo *Nodo**.



La figura 2 se obtiene cuando a partir de la figura 1 se llama:

```
int n= reemplazarNodoK(&a, 4, b);
```

¹ Primero se recorren recursivamente en orden los nodos del subárbol izquierdo, luego el nodo raíz y por último se recorren en orden los nodos del subárbol derecho.

Se reemplazó *u* por *v* y la función retornó 4. La figura 3 se obtiene cuando a partir de la misma figura 1 se llama:

```
int n= reemplazarNodoK(&a, 2, b);
```

Acá se reemplazó la raíz *s* del árbol por *v*. Por eso se requiere que el puntero a la raíz del árbol se pase por referencia (&*a*). Por último si a partir de la figura 1 se intentara reemplazar el quinto nodo de *a* que no existe, no se haría ningún reemplazo y la función retornaría 4.

Ayuda: Sea $a = *pa$, el árbol en donde se hará el reemplazo. Pruebe en orden los siguientes 4 casos. (1) Si *a* es nulo, no se puede hacer ningún reemplazo, retorne 0. Note que *pa* es siempre distinto de nulo. (2) Intente hacer recursivamente el reemplazo en el subárbol izquierdo de *a*. Si tiene éxito retorne *k*. (3) Si el número de nodos del subárbol izquierdo es *k*-1, reemplace el nodo *a* y retorne *k*. (4) Intente hacer recursivamente el reemplazo en el subárbol derecho de *a*, etc.

Pregunta 3

Se necesita programar un sistema cliente/servidor para organizar los vigías del Titanic. El sistema requiere que siempre hayan 2 vigías en busca de icebergs. El servidor se llama *titanic*, corre siempre en *localhost* y escucha a los clientes en el puerto 3000. El cliente se llama *vigia* y recibe como parámetro el nombre de un vigía. El siguiente es un ejemplo de uso del sistema:

<i>Pedro</i>	<i>Juan</i>	<i>Diego</i>	<i>José</i>
\$ vigia pedro pareja: nadie (espera)			
	\$ vigia juan pareja: pedro (espera)		
\$ (termina)		\$ vigia diego pareja: juan (espera)	
	\$ (termina)		\$ vigia jose pareja: diego (espera)

Note que el *prompt* \$ indica cuando un comando termina o si debe esperar. El tiempo avanza hacia abajo. En **negritas** aparece lo que escribió un usuario.

Requerimientos: Programe el cliente y el servidor de este sistema. El cliente debe reproducir exactamente la salida que se muestra en el ejemplo, exceptuando el texto que dice (*espera*) o (*termina*). Cuando llega un nuevo vigía, el vigía que lleva más tiempo vigilando termina. En el servidor use threads para atender los clientes; no programe la función *main*; programe la función de servicio (*serv*). Declare las variables globales que necesite señalando como se inicializan.