

# CC3301 Programación de Software de Sistemas

Control 1 – Semestre Otoño 2014

Prof.: Luis Mateu

## Pregunta 1

Se necesita una función que dado un string  $s$  entregue un nuevo string en donde se han substituido todas las apariciones del string  $p$  por  $r$ . El encabezado de la función es:

```
char *substituir(char *s, char *p, char *r);
```

La siguiente tabla muestra ejemplos de uso:

Invocación	Resultado
<code>substituir("hola mundo", "hello", "bonjour");</code>	"hola mundo"
<code>substituir("hola mundo", "ola", "ello");</code>	"hello mundo"
<code>substituir("elgatotemelperro", "el", "un");</code>	"ungatotemunperro"

- Escriba una función *espacio* que recibe los mismos argumentos que *substituir* y que entrega un entero que corresponde al espacio requerido en bytes para almacenar el resultado de la llamada a *substituir*. (2 puntos)
- Programa la función *substituir*. Ud. debe usar un solo *malloc* para pedir de una vez el espacio requerido por el resultado. Ud. no puede usar otros *malloc*. (4 puntos)

## Pregunta 2

**Parte a.-** Programe eficientemente y usando los operadores de bits la siguiente función:

```
int posicionBits(int x, int p, int n);
```

Esta función busca en  $x$  el patrón  $p$  de  $n$  bits entregando su posición expresada como la posición del bit menos significativo de  $p$  en  $x$ , o -1 si no se encuentra. La siguiente tabla muestra ejemplos de uso de esta función. Los números  $x$  y  $p$  aparecen en binario y no en decimal para facilitar la comprensión de los ejemplos.

Invocación	Resultado
<code>posicionBits(100101, 001, 3);</code>	2
<code>posicionBits(10101001001, 10101, 5)</code>	6
<code>posicionBits(10110, 0, 1)</code>	0

Invocación	Resultado
<code>posicionBits(0, 1, 1)</code>	-1

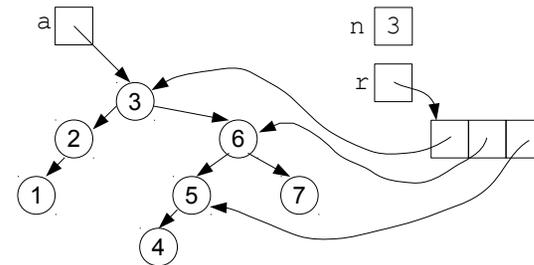
Observe que el parámetro  $p$  es un entero de 32 bits, pero solo se consideran los  $n$  bits menos significativos. Considere además que la posición del bit menos significativo de  $x$  (el de más a la derecha) es 0. La posición del siguiente bit menos significativo es 1. Etc.

**Parte b.-** Programe la función *rama* que dado un árbol de búsqueda binaria  $a$  y un valor  $v$  entregue un arreglo de punteros a los nodos que es necesario visitar para llegar al nodo que contiene  $v$ , o NULL si  $v$  no se encuentra en el árbol. Además debe entregar en  $*p$  el número de nodos en el arreglo. Considere el siguiente ejemplo de uso:

```
typedef struct nodo {
    int v;
    struct nodo *izq, *der;
} Nodo;
Nodo **rama(Nodo *a, int v, int *p);

Nodo *a= ...;
int n;
Nodo **r= rama(a, 5, &n);
```

La siguiente figura muestra cuál debe ser el resultado final de las variables  $n$  y  $r$  cuando se considera que  $a$  es el árbol de la izquierda y se busca el valor 5:



Nota: en los controles Ud. siempre puede programar otras funciones, además de las que se piden en el enunciado. Si hay condiciones de borde que no están especificadas en el enunciado o en los ejemplos de uso, decida Ud. mismo qué hacer (elija el camino en donde la solución resulte más fácil).