

CC3301 Programación de Software de Sistemas – Semestre Primavera 2019 – Tarea 1 – Prof.: Luis Mateu

Parte a.- Programe la función *comprimir* con el siguiente encabezado:

```
typedef unsigned int uint;  
uint comprimir(uint *a, int nbits);
```

Esta función comprime múltiples enteros sin signo almacenados en el arreglo *a* en un solo entero sin signo. Para ello se retorna la concatenación de todos los elementos en *a* truncados a *nbits*. La cantidad de elementos almacenados en el arreglo *a* es el número de enteros de *nbits* que caben en un entero sin signo, es decir el máximo *k* que cumple con $k \cdot nbits \leq \text{sizeof}(\text{uint}) \cdot 8$, en donde $\text{sizeof}(\text{uint}) \cdot 8$ es el tamaño de un entero sin signo (32 bits en la mayoría de las plataformas).

Ejemplo de uso:

```
uint a[] = { 0b 100 110 101 011 000, 0b 000 101 101 011,  
            0b 100 001 010 000 };  
uint r = comprimir(a, 9);  
// r es 0b 101 011 000 101 101 011 001 010 000  
//      <- a[0]  ->  <- a[1]  ->  <- a[2]  ->
```

Observe que al truncar *a[0]* a 9 bits se perdieron los bits más significativos de *a[0]*. Con el fin de facilitar la comprensión se empleó la notación *0b...* para expresar números en base 2, pero no es parte del lenguaje C. Considere que el parámetro *nbits* puede variar entre 1 y $\text{sizeof}(\text{uint}) \cdot 8 - 1$.

Restricción: Ud. no puede usar los operadores de multiplicación, división o módulo (* / %). Use eficientemente los operadores de bits.

Parte b.- Programe la función:

```
char *insercion(char *d, char *s);
```

Esta función entrega un nuevo string en donde se ha insertado el string *s* al principio de *d*. Ninguno de los strings recibidos cambia. Ejemplo:

```
char *s1 = insercion("perro", "gato");  
// s1 es "gatoperro"
```

Restricciones: Use *malloc* para pedir memoria para el string resultante. Sea eficiente: debe pedir exactamente la cantidad de bytes que necesita el resultado, no más. Para el ejemplo debe pedir 10 bytes. Use las funciones *strcpy* y *strcat* para hacer el resto del trabajo.

Parte c.- Programe la función:

```
void insertar(char *d, char *s);
```

Esta función debe insertar eficientemente el string *s* al principio del string *d*. Por lo tanto el string *d* cambia, mientras que *s* no cambia. Ejemplo de

uso:

```
char d[80] = "perro"; // d es un arreglo de 80 bytes con el string "perro"  
insertar(d, "gato"); // d es el string "gatoperro"
```

En el ejemplo, *d* corresponde a un arreglo de 80 bytes, pero solo se ocupan los primeros 6 bytes para almacenar el string "perro" (no olvide el término del string). En el arreglo hay espacio más que suficiente para insertar al principio el string "gato".

Restricciones: Ud. no puede usar el operador de subindicación [], ni su equivalente $*(p+i)$. Para recorrer el string use aritmética de punteros. No puede pedir memoria adicional ni declarar otros arreglos de caracteres, y por lo tanto no puede invocar la función de la parte b. No le servirá usar *strcpy* o *strcat*. Use múltiples punteros para direccionar distintas partes del string.

Sea cuidadoso al desplazar el contenido original de *d* para hacer espacio para *s*. Si copia ascendentemente, el resultado final de *d* en el ejemplo será incorrectamente "gatoperrp". Hágalo descendentemente.

Recursos

Baje *t1.zip* de U-cursos y descomprímalo. El directorio *T1* contiene los archivos *test-t1.c* que prueba si su tarea funciona, *t1.h* que incluye los encabezados de las funciones pedidas y *Makefile*. Ud. debe programar las 3 funciones pedidas en el archivo *t1.c*. El archivo *t1.c.plantilla* muestra algunos includes que Ud. debe agregar en *t1.c*. Lea las instrucciones que aparecen al comienzo de *Makefile*. Ud. debe usar el comando *make* para compilar su tarea. El programa de prueba lo felicitará si su tarea aprueba todos los tests o le indicará cuál test falla.

Entrega

Ud. solo debe entregar el archivo *t1.c* por medio de U-cursos. Entregue su tarea solo si compila sin arrojar warnings en la máquina *anakena.dcc.uchile.cl* y aprueba todos los tests. Se crearán cuentas en *anakena.dcc.uchile.cl* para los alumnos que no son del DCC. Los alumnos del DCC ya tienen cuenta en ese computador. Se descontará medio punto por día de atraso. No se consideran los días sábado, domingo o festivos.