## Algoritmos y Programacion II – Catedra Lic. Gustavo Carolo Evaluacion Parcial – 2015-10-29

– Entregar teoría y práctica por separado –	a	b	С	d	е	f	ø	h	i	Т
Nombre:				-			0			
Mail: Padrón: Padrón:										

## Teoría Tema 1

```
Dado el siguiente código:
            1 #include <stdio.h>
#include <stdlib.h>
                typedef struct {
                   int id_attr
                  char value [17];
               } tInfo;
                typedef struct node {
                  struct node *next;
struct node *prior;
          11
                   char* firstname
                   tInfo attributes[3];
                  union {
    short id_sith;
                      long id_jedi;
                   } number;
                  \mathbf{enum} \ \{ \mathtt{unknown} \!=\! 0, \mathtt{light} \!=\! 1, \mathtt{dark} \!=\! 2 \} \ \mathtt{side} \, ;
                } tNode;
          21 void func(int a, int b, int c) {
                   if (a+b+c > 0) {
                      func (c-2,b-2,a-2);
                      printf("%(\n",a);
                      func(a-1,b-1,c-1);
                      printf("\mathcal{M} \setminus n \mathcal{M} \setminus n",b,a);
          26
                \mathbf{int} \ \mathtt{main} \ (\mathbf{int} \ \mathtt{argc} \,, \ \mathbf{char} {**} \ \mathtt{argv}) \ \{
                   tNode* a:
                  tNode* c
                  a=malloc(sizeof(tNode)); b=malloc(sizeof(tNode)); c=malloc(sizeof(tNode));
                  a->next=b; a->firstname="Luke"; a->prior=c; b->next=c; b->firstname="Obi_Wan"; b->prior=a; c->next=a; c->firstname="Anakin"; c->prior=b;
```

## Asuma que:

41

46

\*a=\*b:

 $\begin{array}{l} \mathtt{func}\left(2\,,3\,,4\right);\\ \mathbf{return}\ 0; \end{array}$ 

a->firstname="Vader";

• Se está compilando en una arquitectura de 64 bits (8 bytes) por palabra.

 $\texttt{printf} \; ("\;\% \_\;\% \_ \; \% \backslash n" \;, \texttt{a->} \texttt{firstname} \;, \texttt{b->} \texttt{firstname} \;, \texttt{c->} \texttt{firstname} \;);$ 

- Un char ocupa 8 bits; un short ocupa 16 bits; un float ocupa 32 bits; el resto de los tipos elementales ocupa 64 bits.
- El compilador ordena las variables en el orden dado y no las serializa (deja espacios en desuso).
- Las variables argc y argv en main no son necesarias.
- La memoria siempre se aloca y malloc nunca devuelve NULL.

## Se pide:

- a. Utilice un diagrama adecuado para justificar el seguimiento recursivo de la función func e indique la salida que imprime.
- b. ¿Cuál es el orden de recursividad de la función func? ¿es directa o indirecta? ¿Cuál es la condición de corte?

- c. Divida con barras los distintos lexemas la función **func** de la misma forma en que lo haría el *scanner*. Indique cual es la cadena que no es procesada como tal por el *scanner*.
- d. El analizador sintáctico define un operador ternario (es decir que recibe tres parámetros), ¿cómo se utliza? ¿conoce operadores unarios?
- e. Complete en bytes las tablas del compilador y la de variables para la función main. Ignore las tablas de constantes y enumerados.
- f. Responda cuántos bytes habría entre a->attributes[10].value[-67] y a->number.id\_jedi. Justifique la cuenta. No analice violaciones de segmentos.
- g. Muestre en un diagrama como quedaran los punteros en memoria. Indique a que parte de la memoria pertenece cada bloque (recuerde incluir todas las variables, incluso aquellas que no se utilicen; argv y argc pueden ser ignoradas). Indique todos los datos que no fueron inicializados. Muestre el resultado que sale al ejecutarse la impresión del printf de la línea 45.
- h. ¿Es posible liberar toda la memoria en el caso anterior? Escriba el código que intente liberar la mayor cantidad posible de memoria (o toda si contesto afirmativamente a la pregunta anterior).
- i. Indicar si en el caso anterior las siguientes expresiones compilan y en caso de hacerlo, un identificador interno o externo tipo:
  - I) \*((\*a).next->firstname)
  - II) &((\*b).number)
  - III) \*(c->prior-7)

$ID_{int}$	$ID_{ext}$	Base	Tamaño	Offset

Cuadro 1: Tabla de variables

$ID_{int}$	$ID_{ext}$	$L_i/L_s$	Elementos	Base	Tamaño	Dimensión

Cuadro 2: Tabla de Arreglos

$ID_{int}$	$ID_{ext}$	$ID_{dat}$	Base	Tamaño	Offset	Total
	Cua	dro 3: 5	Fabla (	de Registi	ros	