

 <b>UCA</b> <small>PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA</small> <b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>Calificación</b>		<b>16 / 07/ 2018</b>	<b>FINAL</b>	<b>TEMA 1</b>
	<b>LEGAJO:</b>		<b>INFORMÁTICA 2 - PROGRAMACIÓN ESTRUCTURADA</b>		<b>Hojas:</b>
	<b>APELLIDO:</b>		<b>NOMBRE:</b>		
<b>Para aprobar PARTE I y PARTE II deberá obtener al menos 27 puntos (sumando ambas partes).</b>					<b>[PARTE I Y II DE III]</b>

**PARTE I - Completar programa (30 puntos completo. 15 puntos cada uno)**

**1 -** Completar las secciones indicadas con “\_\_\_” con una línea de código correspondiente. Complete las secciones indicadas con “\_\_(\*)\_\_” con una o más líneas de código. El resto de las líneas de código del programa deben quedar sin modificar, y en el orden mostrado:

<p><b>1.1) -</b> Completar la función <b>minimoDir</b> para que en forma recursiva retorne la dirección de memoria donde se encuentra el mínimo valor dentro del arreglo pasado por parámetros. El arreglo termina en 0, y nunca se encuentra vacío.</p> <pre>int* minimoDir (int* v) { int *min,_____,_____ ;     x=v;     min=x;     if(*v!=_____){         v++;         m=minimoDir(_____);         if (_____){             min=m;         }     } return _____; }</pre>	<p><b>1.2) -</b> Completar la función <b>bit</b> para que retorne <b>1</b> sólo si el número <b>n</b> (<i>unsigned int</i>) pasado por parámetro tiene <u>los tres bit de la derecha y los tres bit de la izquierda en 1 y el resto de sus bit exactamente en 0</u>. Caso contrario <b>bit</b> deberá retornar <b>0</b>.</p> <pre>int bit (unsigned int n) {     unsigned int mask1=_____ ;     unsigned int mask2=_____ ;     if(n&amp;(mask1 mask2) == _____)         return _____;     else         return _____; }</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**PARTE II - Opción múltiple (32 puntos si todas correctas, 16 puntos si hay 3 correctas, 0 puntos otro caso)**

**2 -** Indicar la opción correcta:

<p><b>2.1) -</b></p> <pre>void sum (int arr[]){     *arr = *arr +3; } void resta (int a){     a= a - 4 ; } int main(){     int a[3]={3,2,13}; int num=1;     sum(a);     resta(num);     printf("%d",*a + num);     return 0; }</pre> <p>¿Que imprime el programa?</p> <p>a) 10    b) 9    c) 2    d) 7 e) 6    f) Hay error en ejecución g) Ninguna de las anteriores</p>	<p><b>2.2) -</b></p> <pre>void f(){     int a=9, b=13;     int* ar[2];     ar[0]=&amp;a;     ar[1]= &amp;b;     printf("%d", *ar[0]+*ar[1]); } int main(){     f();     return 0; }</pre> <p>¿Que imprime el programa?</p> <p>a) 12    b) 13 c) 12    d) 8 e) 22 f) Hay un error en ejecución o compilación g) Ninguna de las anteriores</p>
<p><b>2.3) -</b></p> <pre>int main(){     int* p=NULL;     (*p)=10;     printf("%d",*(p)+1); return 0; }</pre> <p>¿Que imprime el programa?</p> <p>a) 15    b) 10    c) 5    d) 11 e) Hay un error en ejecución o compilación f) Ninguna de los anteriores.</p>	<p><b>2.4) -</b></p> <pre>int foo(int n,int m){     int miFoo=0;     if(m&lt;0){         miFoo=n*foo(n,m-1);         return miFoo;     }     return miFoo; }</pre> <p>¿Qué retorna la función para foo (3,4)?</p> <p>a) 27    b) 12    c) 1    d) 3    e) 81 f) No retorna, hay un loop infinito. g) No retorna. Error en ejecución o compilación. h) Ninguna de las anteriores.</p>