


| | | | | | |
|--|--------------|--|---------------------------|-------|-----------------------|
|  PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias | Calificación | | 13 / 07/ 2017 | FINAL | TEMA 1 |
| | LEGAJO: | | PROGRAMACIÓN ESTRUCTURADA | | Hojas: |
| | APELLIDO: | | NOMBRE: | | |
| Para aprobar PARTE I y PARTE II deberá obtener al menos 27 puntos (sumando ambas partes). | | | | | [PARTE I Y II DE III] |

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

1 - Completar las secciones indicadas con “_____” con una línea de código correspondiente. El resto de las líneas de código del programa deben quedar sin modificar, y en el orden mostrado:

| | |
|--|--|
| <p>1.1) – Completar la función cargarString para que cargue desde teclado un arreglo dinámico de char .(Aclaración: el arreglo debe terminar en '\0'). Un ENTER determina el fin de la carga de datos. <u>Aclaración:</u> Al finalizar la función, en *str debe quedar la dirección de memoria que apunta al inicio del arreglo cargado.</p> <pre>void cargarString (char** str) { int i=0;char aux=""; _____=malloc(sizeof(char)); aux=getche(); while(aux!='\r') { _____ i++; _____ } _____='\0'; }</pre> | <p>1.2) – Completar la función ordenar, que recibe la dirección de un puntero a un arreglo dinámico del tipo t_per. La función deberá ordenar el arreglo en forma descendente por el campo dni.</p> <pre>void ordenar(t_per** per) { int i=0,j=0; t_per aux; for(i=0;_____ ;i++) { for(j=i+1;_____ ;j++) { if(_____) { _____ _____ } } } }</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>2.1) -</p> <pre>int foo(int n,int m){ int miFoo=1; if(m>0){ miFoo=n*foo(n,m-1); } return miFoo; }</pre> <p>¿Qué retorna la función para foo (3,4)?</p> <p>a) 27 b) 12 c) 90 d) 3 e) 4</p> <p>f) No retorna, hay un loop infinito.</p> <p>g) No retorna. Error en ejecución o compilación.</p> <p>h) Ninguna de las anteriores.</p> | <p>2.2) -</p> <pre>int main() { char* s[5]; int i=0; char c='2'; s[i]=&c; for(i=0;i<5;i++){ s[i]=&c; c=c+1; } printf("(%d, %d)", (c=='2'), !(s[0]==&c)); return 0; }</pre> <p>¿Qué imprime al ejecutar el código ?</p> <p>a) (0,0) b) (0, 1) c) (1, 1) d) (1, 0)</p> <p>e) No imprime. Error en ejecución o compilación.</p> <p>f) Ninguna de las anteriores.</p> |
| <p>2.3) -</p> <pre>int main(){ char* s=NULL; int i; for(i=0;i<4;i++){ *(s+i)='a'+i; } s[i]='\0'; printf("%s",s); return 0; }</pre> <p>¿Qué imprime el programa?</p> <p>a) a b) ab c) abcd d) 01234 e) 0</p> <p>f) No imprime. Error en ejecución o compilación.</p> <p>g) Ninguna de las anteriores</p> | <p>2.4) -</p> <pre>int foo(unsigned char n,unsigned char filter){ return n&((255&(filter<n+n))>1); }</pre> <p>¿Qué retorna la función para foo (3,4)?</p> <p>a) 255 b) 128 c) 4 d) 3 e) 0</p> <p>f) No retorna. Error en ejecución o compilación.</p> <p>g) Ninguna de las anteriores.</p> |