

PARCIAL 2 (25%)

PARTE I: Seleccione exactamente una alternativa correcta en las siguientes preguntas. **(5pto)**

<p>1) ¿Qué muestra en pantalla la sentencia printf?</p> <pre>int f(int y){ return (y*2 + 1);} void main() { int x = 10, y = 1; y = f(2*x); printf ("%d, %d", x,y);}</pre>	<p>____ (a) 41,10 ____ (b) 10, 1 ____ (c) 20, 1 ____ (d) 10, 41</p>
<p>2) ¿Qué muestra en pantalla la sentencia printf?</p> <pre>int x = 3, *y; y = &x; x = 5; *y = 2; printf("%d",x);</pre>	<p>____ (a) 3 ____ (b) 5 ____ (c) 2 ____ (d) 1</p>
<p>3) ¿Qué muestra en pantalla la sentencia printf?</p> <pre>int f(int *y){ *y = 20; return 1;} void main() { int x = 0, y = 1; x = f(&y); printf ("%d, %d", x,y); }</pre>	<p>____ (a) 0, 1 ____ (b) 20, 1 ____ (c) 1, 20 ____ (d) 1, 0</p>
<p>4) ¿Qué muestra en pantalla la sentencia printf?</p> <pre>int i = 3; while (i--) { int i = 100; i--; printf("%d ", i); }</pre>	<p>____ 99 99 99 ____ 99 98 97 ____ 2 2 2 ____ ciclo infinito</p>

PARTE II: Convierte las siguientes estructuras de control a *do ... while*

(2pto)

<pre>for(i=1; i<10; i++){ if (i%3==0) continue; printf("%i", i); }</pre>	
<pre>count = 0; while (count<10 && (count+= 1)) printf("count is %d\n", count);</pre>	

PARTE III: Complete el siguiente programa**(8pto)**

```
#include <stdio.h>
/* La función minSerie recibe por teclado una serie de números enteros hasta que se encuentra un
número mayor a (100) y determina el menor de los números (no negativos) insertados y la posición en
que se inserto y retorna el número de elementos (no negativos) de la serie*/

int minSerie(int _____, int _____)
{
    int x; // variable de entrada de elemento de la serie
    int n = 0; // cantidad de elementos leídos de la serie
    scanf("%d",&x); // Lee el primer elemento

    _____ ; // Inicializa el valor menor

    _____ ; // Inicializa la posicion del menor

    _____ // El comienzo del ciclo
    {
        n = n+1; // se leyó un elemento más de la serie
        _____ // x es menor que los anteriores
        {
            _____ ; // Actualiza el valor menor

            _____ ; // Actualiza la posición del menor
        }
        scanf("%d",&x); // Lee el siguiente elemento
    }
    return _____; // retorna el numero de elementos de la serie
}

int main()
{
    int cant; // cantidad de elementos en la secuencia
    int xMin; // elemento menor de la secuencia
    int pMin; // posición en la secuencia del elemento menor
    printf("Inserte una serie de valores enteros.\n");
    printf("(termine con un número mayor a 100)\n");

    cant = minSerie(_____, _____);
    if (cant>0){
        printf("La serie leida fue de %d elementos.\n", cant);
        printf("El menor de los numeros fue %d.\n", xMin);
        printf("se inserto en la posicion %d.\n", pMin);
    } else {
        printf("No se inserto ningún número mayor o igual a
        cero!\n");
    }
    return 0;
}
```

PART IV : Se desea escribir un programa que realice las tareas que se indican a continuación siguiendo las restricciones que se dan : **(10pts)**

- a) Leer de la entrada estándar dos enteros positivos a y b tales que $a < b$ el programa no debe continuar hasta que esta condición se satisfaga.
- b) Luego leer una secuencia de números hasta que aparezca uno negativo y
 - 1.- Contar cuántos de estos números están en el intervalo semi-abierto $[a; b)$,
 - 2.- Hallar el promedio de los que están en dicho intervalo.
- c) El programa debe constar del main y dos sub-programas como se indica abajo.
 - void leerIntervalo(int *a, int *b)
 - float promedio(int a, int b)
- d) Ambos sub-programas deben usar un ciclo do-while.

Se deben especificar las entradas, pre-condiciones, salidas y post-condiciones del programa principal y de cada uno de los subprogramas.