

**PARCIAL 2 (25pts)**

**PARTE 1:** Seleccione exactamente una alternativa correcta en las siguientes preguntas. **(5 pts)**

1) Cuando una función que retorna un entero es invocada, se ejecutan las acciones que la definen y luego se devuelve el control a:	____ (a) La instrucción inmediatamente después del main() ____ (b) La instrucción donde se hizo la llamada ____ (c) Al siguiente procedimiento o función ____ (d) La instrucción siguiente donde se hizo la llamada.
2) Dado el siguiente fragmento de programa: Que imprimirá la sentencia printf? <pre>int *a, x, y; int v[4]={1, 2, 3, 4}; x=1; a=&amp;x; y=2; x=y*2; printf("%d %d", y, *a);</pre>	____ (a) 2 1 ____ (b) 2 0 ____ (c) 2 4 ____ (d) 2 2
3) Dada las siguientes inicializaciones de cadenas de caracteres en lenguaje C, diga cuál es la correcta:	____ (a) char *Cad1 = { 'Argentina\0' }; ____ (b) char *Cad1 = "Argentina"; ____ (c) char *Cad1 = { "Argentina\0" }; ____ (d) char Cad1 = "Argentina";
4) Diga cuántas veces se realiza la instrucción: prod=i+j; en el siguiente trozo de programa: <pre>for (i=1; i&lt;=6; i++ )     for (j=10; j&gt;= 1; j-- )         prod=i+j; printf(" \n i= %d j= %d prod= %d\n", i,j,prod);</pre>	____ (a) 60 veces ____ (b) 55 veces ____ (c) 66 veces ____ (d) ninguna de las anteriores
5) Diga cuál de los trozos de código calcula el promedio del arreglo A de valores reales:  <pre>float pro, acm=0; int i; float A[]={4.0, 5.0, 2.0, 3.0, 1.0, 5.0, 5.0, 6.0};</pre>	____ (a) for (i = 0; i <= 8; i++) acm = acm + A[i]; pro = acm / i; ____ (b) for (i = 0; i < 8; i++) acm = acm + A[i]; pro = acm / i; ____ (c) for (i = 0; i < 8; i++) acm = acm + A[i]; pro = acm / (i+1); ____ (d) for (i = 0; i < 7; i++) acm = acm + A[i]; pro = acm / i;

**PARTE 2:** Realice la corrida en frío de los siguientes programas (6 pts)

<pre>#include&lt;stdio.h&gt;  int global_1 = 7, global_2 = 21, global_3 = 35;  void funcion_a(void); void funcion_b(void); void funcion_c(int global_2, int *local_b_2);  int main(int argc, char * argv[]) {     int local_main_1 = 11, local_main_2 = 22;     int global_1 = 111;     funcion_a();     funcion_b();     printf("\nValores en main():\n\n");     printf("local_main_1 = %d\n", local_main_1);     printf("local_main_2 = %d\n", local_main_2);     printf("global_1 = %d\n", global_1);     printf("global_2 = %d\n", global_2);     printf("global_3 = %d\n", global_3); }  void funcion_a(void) {     int local_a_1 = 111, local_a_2 = 222;     int global_2 = 222;     printf("\nValores en funcion_a():\n\n");     printf("local_a_1 = %d\n", local_a_1);     printf("local_a_2 = %d\n", local_a_2);     printf("global_1 = %d\n", global_1);     printf("global_2 = %d\n", global_2);     printf("global_3 = %d\n", global_3); }  void funcion_b(void) {     int local_b_1 = 1111, local_b_2 = 2222;     int global_3 = 333;     printf("\nValores en funcion_b():\n\n");     printf("local_b_1 = %d\n", local_b_1);     printf("local_b_2 = %d\n", local_b_2);     printf("global_1 = %d\n", global_1);     printf("global_2 = %d\n", global_2);     printf("global_3 = %d\n", global_3);     funcion_c(global_2, &amp;local_b_2); }  void funcion_c(int global_2, int *local_b_2) {     int local_c_1 = 7;global_1= 5;     printf("\nValores en funcion_c():\n\n");     printf("global_2 = %d\n", global_2);     printf("local_b_2 = %d\n", *local_b_2);     printf("global_1 = %d\n", local_c_1);     printf("global_2 = %d\n", global_1);     global_2 = 5; *local_b_2 = 6; } </pre>	<p><b>Memoria Global</b></p> <hr/> <p><b>Memoria Main</b></p> <hr/> <p><b>Memoria</b> funcion_a</p> <hr/> <p><b>Memoria</b> funcion_b</p> <hr/> <p><b>Memoria</b> funcion_c</p> <hr/> <p><b>Pantalla</b></p>
--	--

**PARTE 3:** Complete el siguiente programa en C, el cual lee un número y un arreglo de N elementos y determina si el número se encuentra o no en el arreglo; en caso afirmativo indica la posición (índice) donde se encuentra. ( 6 pts)

*/\* Este programa recibe un arreglo no vacío de N elementos y un número x y determina si el número x aparece o no en el arreglo, y en caso afirmativo, encuentra la posición del arreglo donde aparece el número x. \*/*

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0
#define max 100
int main()
{
    int A[max], N; /*Entrada: N=número de elementos, A= arreglo de N elementos*/
    int x; /*Entrada: número proporcionado por el usuario.*/
    /* Precondicion: 0<N<=max */
    int esta, pos; /* Salida */
    /* Poscondicion: esta = TRUE si el elemento x se encuentra en el arreglo, FALSE en caso contrario. Si esta = TRUE entonces A[pos] = x */
    int i,j; // Declaración de variables auxiliares (en caso de necesitarlas)
    /* Escriba las instrucciones necesarias para leer y validar el valor de N y los elementos del arreglo A */
```

```
printf (“Introduzca el valor del elemento a buscar en el arreglo: “);
scanf (“%d”, &x);
```

```
/* Escriba las instrucciones necesarias para determinar si x se encuentra en el arreglo y en caso afirmativo, en que posición (índice) */
```

```
    esta = _____
    i=0;
    while ( _____ )
    {

    }

    if ( _____ )
        printf( “el elemento %d se encuentra en la poscion %d \n”, _____ );
    else
        printf (“ el elemento %d NO se encuentra en el arreglo \n”, _____ );
```

```
}
```

**PARTE 4:** Escriba un programa para calcular el producto del factorial de tres (3) números enteros leídos por pantalla. El programa debe tener el programa principal main y utilizar las siguientes funciones ( **8 pts** ) :

- int LeerEntero ()
- factorial (int n, int \*f)