

Alumno: _____ Legajo: _____ Nota _____

Condiciones de entrega:

- a. La hora de inicio es 19.00 y la de finalización es 20.30
- b. Todas las entregas realizadas fuera de término no serán tenidas en cuenta para su corrección
- c. Condición de aprobación 6 o más puntos

Parte teórica:

1. Dado el string definido como char v[]="Hola" indique la opción verdadera
 - a. El strlen (v) es 5
 - b. El strlen (v) es 4
 - c. El strlen (v) es 3
 - d. Ninguna de las anteriores
2. Indique cuál de las siguientes sentencias es correcta
 - a. int r = 10 % 2;
 - b. float r = 2.7 % 2;
 - c. float r = 10 % 2.0
 - d. Todas las anteriores son correctas
3. Dada la siguiente variable int var = 0x1234; que imprime el siguiente printf("0x%lx\n", ~(var ^ 0xFFFF))
 - a. 0xFFFF
 - b. 0x1234
 - c. 0xEDCB
 - d. Ninguna de las anteriores
4. Dado el siguiente código indique qué proposición es verdadera char v[10]; int n = strlen (v);
 - a. n vale 1
 - b. n vale 10
 - c. Genera una violacion de segmento
 - d. Hay un error de sintaxis.

Parte práctica:

1. Implemente una función que reciba un vector de números enteros y realice dos operaciones sobre él.
 - a. **Insertar al final:** Elimina el primer elemento del vector, desplaza todo un lugar para dejar el último lugar libre e inserta el nuevo valor al final. Ejemplo: Inserta el número 33 en un vector de 4 posiciones.
Vector original.

11	12	13	14
----	----	----	----

Vector luego de insertar el valor.

12	13	14	33
----	----	----	----

- b. **Insertar al inicio:** Elimina el último elemento del vector, desplaza todo un lugar para dejar el primer lugar libre e inserta el nuevo valor al inicio. Ejemplo: Inserta el número 33 en un vector de 4 posiciones.
Vector original.

11	12	13	14
----	----	----	----

Vector luego de insertar el valor.

33	11	12	13
----	----	----	----

El prototipo de la función es el siguiente:

```
int insertarInt(int *dataPtr, int dataCant, int n, int op)
```

Donde:

- dataPtr: Puntero al vector con números enteros.
- dataCant: Cantidad de elementos del vector de enteros.
- n: Numero a insertar
- op: Operación a realizar.
 - 0: Inserta el inicio.
 - 1: Inserta al final.

Devuelve el valor eliminado del vector.

2. Implemente una función que divida un archivo en varios archivos del tamaño indicado. El prototipo de la función es el siguiente

```
int dividirArchivos(char *nombre, int tamano)
```

Donde:

- nombre: Nombre del archivo a dividir.
- tamano: Tamaño de cada archivo resultante de la división.

Devuelve: La cantidad de archivos en los que se dividió el original o un número negativo indicando el error.

- -1: Si no pudo abrir el archivo origen
- -2: Si la cantidad de archivos resultantes es mayor que 999
- -3: Si la cantidad de archivos resultantes es cero

El nombre de los archivos resultantes serán numerados luego de la extensión, por ejemplo si el nombre del archivo original es prueba.txt y este se dividirá en 4 partes, los nombres de los archivos resultantes serán los siguientes prueba.txt.000; prueba.txt.001; prueba.txt.002; prueba.txt.003;

Solución parte teórica:

1. Opción B
2. Opción A
3. Opción B
4. Opción C

Solución parte práctica:

```
int insertarInt(int *dataPtr, int dataCant, int n, int op) {  
    int i, valor;  
    if (op == 0) {  
        valor = *(dataPtr + dataCant - 1);  
        for (i = dataCant - 1; i > 0; i--) {  
            *(dataPtr + i) = *(dataPtr + i - 1);  
        }  
        *(dataPtr + 0) = n;  
    } else {  
        valor = *(dataPtr + 0);  
        for (i = 0; i < dataCant - 1; i++) {  
            *(dataPtr + i) = *(dataPtr + i + 1);  
        }  
        *(dataPtr + dataCant - 1) = n;  
    }  
    return (valor);  
}
```

```
int dividirArchivos(char *nombre, int tamano)
{
FILE *fr, *fw;
int rVal, tamanoTotal;
struct stat data;
char nombreParte[64];
char *p;
int tamanoLeido, i, j;

    fr = fopen (nombre, "r");
    if (fr == NULL) {
        return (-1);
    }

    rVal = stat(nombre, &data);
    if (rVal < 0) {
        return (-1);
    }
    tamanoTotal = data.st_size;

    p = (char *) malloc (sizeof (int) * tamano);
    if (p == NULL) {
        return (-4);
    }

    j = 0;
    do {

        tamanoLeido = fread (p, 1, tamano, fr);
        if (tamanoLeido > 0) {
            sprintf (nombreParte, "%s.%03d", nombre, j);
            fw = fopen (nombreParte, "w");
            if (fw == NULL) {
                free (p);
                fclose (fr);
                return (-2);
            }

            fwrite (p, 1, tamanoLeido, fw);
            fclose (fw);
            printf ("%s. %d. %d\r\n", nombreParte, tamanoLeido, j);
            j++;
        }
    } while (tamanoLeido > 0);
```

```
free (p);
fclose (fr);

return (0);
}
```