

Clase de Repaso – Módulo 2

Taller de Lenguajes

Ejercicio 1

- Dado el siguiente código, indique y corrija los errores para que funcione correctamente.

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define n 10
```

```
int factorial(int n){
    if (n <= 1)
        return 1;
    return n * factorial(n - 1);
}
```

```
int main(){
    . . .
    printf("el factorial de %d es %d", n, factorial(n));
    return 0;
}
```

se debe reemplazar la n del define por otra y usarla en el main unicamente

Ejercicio 2

Defina una macro que permita calcular la suma de los cubos de 2 números

```
#define SUMA_CUBOS(a,b)
#include <stdio.h>
#include <stdlib.h>
#define SUMA_CUBOS(a,b) ((a)*(a)*(a)+(b)*(b)*(b))

int main(){
    int x,y,z;
    printf("Ingrese un numero: ");
    scanf("%d",&x);

    printf("Ingrese otro numero: ");
    scanf("%d",&y);

    z = SUMA_CUBOS(x,y);

    printf("La suma de los cubos es: %d", z);

    return 0;
```

Ejercicio 3

Indique qué imprime el código siguiente:

```
#define IMPRIME 0
```

```
int main() {  
    int i;  
    for (i=0; i<5; i++){  
        #ifdef IMPRIME  
            printf("%d ", i);  
        #undef IMPRIME  
        #else  
            printf("%d ", 4-i);  
        #endif  
    }  
    return 0;  
}
```

imrpime: 0-1-2-3-4

Ejercicio 4

Para cada inciso indique si es verdadero o falso.

- | | |
|---|---|
| F | a) Si sólo se va a leer un archivo binario es lo mismo abrirlo con modo "ab" que con "a+b". |
| V | b) $p = \text{realloc}(p, s)$ es equivalente a $p = \text{malloc}(s)$ siempre y cuando p sea nulo. |
| V | c) <code>rb+</code> o <code>r+b</code> se pueden utilizar indistintamente para abrir un archivo binario para leer y escribir. |
| F | d) Solo es posible crear un tensor de enteros de manera dinámica si se declara <code>int ***</code> . |
| F | e) Una variable asociada a un archivo de texto se declara con <code>FILETEXT *</code> . |

Ejercicio 5

Escriba un programa que lea un número entero desde teclado (n) y reserve memoria en forma dinámica para una matriz de n filas por n columnas. Inicialice los valores de la matriz a partir de valores ingresados desde teclado. A continuación, imprima la suma de las 2 diagonales principales. Por último, libere la memoria reservada.

Ejemplo: $n = 3 \rightarrow$

1	4	5
0	2	3
5	8	0

Debe imprimir:

Diagonal 1 $\rightarrow 3$

Diagonal 2 $\rightarrow 12$

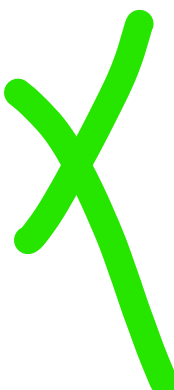
Nota: modularice la reserva de memoria, la inicialización, el cálculo e impresión y la liberación de memoria.

Ejercicio 6

El archivo de texto “**recaudacion.txt**” contiene información de los tickets correspondientes a las cajas de una juguetería. En cada línea del archivo se encuentra: número de ticket, medio de pago (Efectivo, Tarjeta de Débito o Tarjeta de Crédito). Estos valores se encuentran separados por un único blanco tal como se observa en el cuadro que aparece a derecha.

Escriba un programa que lea la información del archivo “**recaudacion.txt**” y la utilice para generar el archivo binario “**recaudacion.dat**” con la siguiente estructura:

```
typedef struct {  
    int  nroTicket;  
    char medioPago;  
    float monto;  
} Recaudacion;
```



1	E	1200.35
2	D	508.20
3	C	3000.00
5	E	232.43
6	E	1107.12
7	D	580.45
8	D	400.00
10	E	300.12
11	C	103.44
12	D	2281.30
...

Al finalizar deberá mostrar el monto total recaudado con tarjeta de crédito.

Ejercicio 7

Escriba un programa que reciba una operación de multiplicación, división, resta o suma (carácter `x`, `/`, `-` o `+` respectivamente) y dos números enteros como argumentos a la función *main* e imprima el resultado de la misma. En caso de que el programa no reciba los 2 números y la operación correspondiente se debe imprimir un mensaje de error.