



Apellido, Nombre:

.....

Mail:

.....

Padrón:

.....

Teórico / Práctico - Entrego hojas

Nota Final

1:

2:

3:

4:

.....

Aclaraciones:

- Antes de comenzar a resolver el parcial, complete sus datos en esta hoja, y al finalizarlo, firme todas las hojas.
- Se deben numerar TODAS las hojas e inicializarlas con nombre, apellido, padrón.
- La aprobación del parcial está sujeta a la correcta realización de al menos el 50% de 3 ejercicios y sumar 10 puntos en total como mínimo.

1) Aritmética de punteros / Memoria Dinámica (5 puntos)

- a. Dado el siguiente algoritmo, determinar que se imprime. ¿Cuánta memoria se pierde? ¿Cómo la liberaría?

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

const size_t MAX_ANCHO = 5;
const size_t MAX_ALTO = 5;

int main(){
    int*** matriz = malloc(MAX_ALTO*sizeof(int**));
    for (int i = 0; i < MAX_ALTO; i++){
        matriz[i] = malloc(MAX_ANCHO*sizeof(int*));

        for (int i = 0; i < MAX_ALTO; i++){
            int* numero = malloc(sizeof(int));
            for (int j = 0; j < MAX_ANCHO; j++){
                *numero = i+j;
                matriz[i][j] = numero;
                *numero = i*j;
            }
        }

        for (int i = 0; i < MAX_ANCHO; i++){
            for (int j = 0; j < MAX_ALTO; j++){
                printf("%i\t", *(matriz[i][j]));
            }
            printf("\n");
        }

        free(matriz);
        return 0;
    }
}
```

"En una sociedad que nos educa para la vergüenza, el orgullo es una respuesta política." - Carlos Jáuregui.

b. ¿Cuánto vale **a** al momento de imprimirse por pantalla?

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

int main(){
    int a = 4;
    int* b;
    int* c = malloc(sizeof(int));
    int** d = &c;

    b = *d;
    **d = 8;
    a = *c;
    **d = *b;
    *b = 16;

    printf("a vale: %i\n", a);

    return 0;
}
```

2) Análisis de Algoritmos (5 puntos)

¿Cuál de estos dos algoritmos tiene menor complejidad computacional?. Explicar detalladamente el por qué:

```
bool es_unico( int vector[], size_t tope){
    int i,j;
    bool unico=true;

    for (i=0; i < tope ; i++)
        for (j=i+1; j < tope; j++)
            if (vector[i]==vector[j]) unico = false;
    return unico;
}

bool es_unico( int vector[], size_t tope){
    int i,j;
    int ordenado[];
    bool unico=true;

    ordenar(vector,tope,ordenado);
    for (i=0; i < tope-1 ; i++)
        if (ordenado[i]==ordenado[i+1]) unico = false;
    return unico;
}
```

Nota: el método de ordenamiento tiene un orden de ejecución **$O(n\log n)$** .

"En una sociedad que nos educa para la vergüenza, el orgullo es una respuesta política." - Carlos Jáuregui.

3) TDA (5 puntos)

Pajaritos es el nombre que recibe la red de espías de Varys en los Siete Reinos, y muy probablemente en el continente de Essos. Los pajaritos usualmente proporcionan a Varys información muy valiosa. Cada pajarito lleva información de un lugar de los siete reinos a Lord Varys, esta información es un mensaje, que normalmente proviene de alguna fuente de datos. Lord Varys posee una lista de cada uno de sus pajaritos con cada mensaje enviado, ya que los mantiene ordenados por fecha.

1. Definir el tda pajaritos (struct y firmas).
2. Implementar las siguientes primitivas:

```
/*
 * Dará de alta un pajarito, ampliando la memoria reservada.
 * Devuelve 0 si pudo o -1 si no pudo o -2 si ya existe el pajarito.
 */
int varys_nuevo_pajarito(pajaritos_t* pajaritos, char* nombre, char* ciudad);

/*
 * Se insertará un mensaje enviado por un determinado pajarito
 *
 * Devolverá 0 si pudo insertar el mensaje -1 si no,
 * si existe, devolverá -2.
 */
int varys_recibir_mensaje(pajaritos_t* pajaritos, char* nombre, mensaje_t* mensaje);
```

4) Recursividad (5 puntos)

Completar la función **imprimir_raro_n** de forma tal que imprima los caracteres en posiciones impares del string (primero, tercero, quinto, etc) hasta llegar al final y luego imprima los caracteres pares pero empezando por el final. Adicionalmente, el algoritmo debe finalizar inmediatamente si el próximo carácter a imprimir es ':'.

Por ejemplo, para el string **A2lsgoomrti** se debe imprimir **Algoritmos2** y para el string **Aopnr:oobda** se debe imprimir **Aprobado**.

```
#include <stdio.h>

int imprimir_raro_n(const char* string, int n){
}

void imprimir_raro(const char* string){
    imprimir_raro_n(string, 1);
}

int main(){

    imprimir_raro("A2lsgoomrti");

    printf(" ");

    imprimir_raro("Aopnr:oobda");
}
```