



Nombre: \_\_\_\_\_ Grupo 102- \_\_\_\_\_

Examen Final Departamental de Programación Estructurada - Ingeniería en Computación

Fecha: 10/feb/2017

**Problema 1:** Memoria dinámica / Recursividad / Pruebas de escritorio / representación de los datos en memoria / Paso de parámetros a funciones  
Suponga el siguiente fragmento de código:

```
int funcion2(char *p[], int c){
    if(*p==NULL)
        return c;
    return funcion2(p+1, c+1);
}

char **funcion(char *p[], int *n){
    int a=funcion2(p, 0);
    char **r = (char **)malloc(sizeof(char*)*a);
    int pos=0;

    while(*(p+pos)!=NULL) {
        *(r+pos)=(char *)malloc(sizeof(char)*(strlen(*(p+pos))+1));
        strcpy(*(r+pos), *(p+pos));    pos++;
    }
    return r;
}

void main(){
    char *p[10]={"uno", "trece", "cinco"};
    char **w=NULL;
    int n;
    w=funcion(p, &n);
}
```

Dibuje un diagrama donde muestre la memoria después de ejecutar la última línea de la función main → indique qué tipo de apuntadores y hacia donde apuntan, los valores que hay en cada localidad de memoria, las direcciones de memoria. Suponga que se trabaja con direcciones de 4 bytes.

(valor: 2 puntos)

**Problema 2:** Recursividad

Escriba una función recursiva que regrese la cantidad de vocales (mayúsculas/minúsculas) que existen en una cadena recibida como parámetro.

(valor: 3 puntos)

**Problema 3:** Estructuras / métodos de ordenamiento / archivos

Se deben almacenar registros de personas (P) en un archivo. Los datos que deben almacenarse son: el nombre o nombres de la persona, apellidos, fecha de nacimiento y número de seguro social.

- Declare la fecha (F) en una estructura y una estructura de tipo persona que contenga los datos indicados, pero haciendo uso del campo de fecha. Escriba una función que reciba una persona y muestre sus datos (+0.5).
- Escriba una función que reciba un vector de personas, el número de personas y los guarde en un archivo binario cuyo nombre se recibirá como parámetro. Abra el archivo borrando el contenido, en caso de existir (+1.0).
- Escriba una función que reciba un vector de personas, número de personas y lo ordene en base a la fecha de nacimiento, dejando al inicio a las personas más jóvenes (+1.0).
- Escriba una función que reciba el registro de una persona (P) y una fecha (F). Suponga que la fecha es posterior al nacimiento de la persona. La función debe regresar un entero que corresponde a la edad, en años de dicha persona (+1.0).

(total: 3.5 puntos)

**Problema 4:** Apuntadores/funciones/memoria dinámica/estructuras/entrada y salida con formato

Complete el siguiente código, basado en los comentarios y estructuras definidas al inicio.

(total: 3 puntos)

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

typedef struct materia{
    char *nombre;
    float par1, par2, par3, ordinario;
}M;

typedef struct estudiante{
    int matricula;
    char *nombre;
    int numMats;
    M *mats;
} E;
```

```

void leerMateria(M *m){ //leer los datos de una materia recibida por referencia
    char tmp[100];
    gets(tmp);
    //solicitar memoria para almacenar el nombre de la materia en la estructura
    m->nombre = (_____)malloc( sizeof(char) * ( _____ + _____ ));
    strcpy(m->nombre, tmp);
    //pedir los datos de cada una de las tres materias (usar el operador flecha)
    scanf("%f%f%f%f\n", _____, _____, _____, _____);
}

void mostrarDatos(E e){ //Mostrar los datos de un estudiante
    int i;
    printf("Estudiante %-10s\n", _____ );
    printf("Matricula   %10d\n", _____);

    for(i=0;i<e.numMats; i++){
        printf("Materia: %-15s\n", e.mats[ ____ ].nombre);
        printf("Calificaciones:  %8.1f%8.1f%8.1f%8.1f  => ",
            _____, _____, _____, _____);
        printf("%8.1f\n\n", ((_____+ _____+ _____)+ 3.0*_____)/_____);
    }
}

int main(){
    char linea[100];          E persona;          int i;
    //Leer la matricula del estudiante
    scanf("%d\n", _____);
    //pedir el nombre del estudiante
    gets(linea);

    //solicitar memoria para almacear el nombre del estidiante.
    persona._____ = (_____)malloc( sizeof(_____) * (_____));
    //copiar el nombre en el miembro correspondiente en persona.
    strcpy(_____._____, linea);

    //solicitar el número de materias que lleva el estudiante
    scanf("%d\n", _____);
    persona.mats = (_____)malloc( sizeof(____) * _____);
    if(persona.mats==_____){ //validación
        printf("Error no hay memoria\n");      exit(0);
    }
    //leer cada materia
    for(i=0; i<persona._____ ; i++)
        leerMateria( _____.mats[i] );
    //Mostrar los datos leídos
    mostrarDatos(_____);
}

```