

NOMBRE: _____

APELLIDOS: _____

D.N.I.: _____

CALIFICACIÓN:

NORMAS:

- Sólo se pueden usar 2 bolígrafos y un corrector. Es lo único que tiene que haber en la mesa.
- Sólo se corregirá lo que esté escrito a bolígrafo.
- No se puede consultar el móvil ni ningún otro dispositivo durante el examen.
- No se puede utilizar material de consulta ni calculadora.
- No se puede hablar durante el examen.
- Se puede escribir en la hoja del examen por ambas caras, así como en las hojas de examen.
- Se debe poner el nombre, apellidos y DNI en todas las hojas.
- Se deben entregar todas las hojas de examen, incluidas las hojas usadas como borrador.

1. **(30 puntos)** Realice un programa en lenguaje C con las características siguientes:

- a. La función `main` deberá leer una cadena de como máximo 100 caracteres y un entero que indicará cuántas letras se quieren contar. Hay que comprobar que dicho entero sea positivo. Si no, dará un mensaje de error y se volverá a leer hasta que lo sea. A continuación, se leerán tantas letras como el entero leído indique, guardándolas en un vector. Empleando la función `cuenta` se contarán cuántos caracteres iguales a los leídos tiene la cadena leída. La función `main` mostrará el resultado por pantalla.
- b. La función `cuenta` recibe una cadena, un vector de letras, un vector de números y el tamaño del vector (es el mismo en ambos vectores) y escribe en cada posición del vector de enteros cuántas letras hay de ese tipo en la cadena pasada como argumento, de forma que el elemento *i*-ésimo del vector de enteros indicará cuantas veces está la letra *i*-ésima del vector de letras en la cadena. No imprime nada por pantalla y no devuelve nada.
- c. Ejemplo de funcionamiento: si se lee el número 4, y la cadena `Cadena de prueba` y las letras `a, e, d y p` el resultado sería: "hay 3 a, hay 3 e, hay 2 d, hay 1 p".

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

```
void cuenta(char cad[100], int *v, char *l, int n);
int main(){
    char c[100], *l;
    int *v, n, i;
    printf("Introduce cadena:");
    fgets(c,100,stdin);
    do{
        printf("\nIntroduce número: ");
        scanf("%d", &n);
        if (n<=0)
```

```

        printf("\nError");
    }while(n<=0);
    l=(char*)malloc(n*sizeof(char));
    v=(int*)malloc(n*sizeof(int));
    printf("\nIntroduce %d letras separadas por intro: ", n);
    while(getchar()!='\n');
    for(i=0;i<n;i++){
        scanf("%c", l+i);
        while(getchar()!='\n');
    }
    cuenta(c, v, l, n);
    printf("\nEl número de letras es: \n");
    for(i=0;i<n;i++)
        printf("Hay %d letras %c\n", v[i], l[i]);
    free(l);
    free(v);
    return 0;
}

void cuenta(char cad[100], int *v, char *l, int n){
    int i, j=0;
    for(i=0;i<n;i++){
        j=0;
        v[i]=0;
        while(cad[j]!='\0'){
            if(cad[j]==l[i])
                v[i]++;
            j++;
        }
    }
    return;
}

```

2. **(15 puntos)** Escriba una función en lenguaje C que dada una matriz de enteros A 100x100 y 2 enteros i y j cuyo valor está entre 0 y 99, intercambie la fila i con la fila j de la matriz, modificando la propia matriz A. No devuelve nada ni imprime nada. Por ejemplo, dada la matriz siguiente y los números 1 y 0, la matriz resultante será la indicada:

$$\text{Matriz} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,100} \\ a_{2,1} & a_{2,2} & \dots & a_{2,100} \\ \vdots & \vdots & \vdots & \vdots \\ a_{100,1} & a_{100,2} & \dots & a_{100,100} \end{pmatrix} \quad \text{Matriz resultante} = \begin{pmatrix} a_{2,1} & a_{2,2} & \dots & a_{2,100} \\ a_{1,1} & a_{1,2} & \dots & a_{1,100} \\ \vdots & \vdots & \vdots & \vdots \\ a_{100,1} & a_{100,2} & \dots & a_{100,100} \end{pmatrix}$$

```

void intercambia(int m[100][100], int i, int j){
    int t, aux;
    for(t=0;t<100;t++){
        aux=m[i][t];
        m[i][t]=m[j][t];
        m[j][t]=aux;
    }
    return;
}

```

```
}
```

3. **(15 puntos)** Escriba una función en lenguaje C que recibe como argumento un vector y su tamaño y lo inicializa con enteros aleatorios no repetidos entre 1 y 50. No devuelve nada.

```
void inicializa(int*v, int n){
    int i,j,aux,repetido=0;
    for(i=0;i<n;i++){
        aux=rand()%50 +1;
        repetido=0;
        for(j=0;j<i;j++){
            if(v[j]==aux){
                repetido=1;
                j=i;
            }
        }
        if(repetido==0)
            v[i]=aux;
        else
            i--;
    }
    return;
}
```

4. **(6 puntos)** Considerando la siguiente estructura, completa el código para leer los datos para todos los coches (num_coches) y almacenarlos en los campos correspondientes:

```
struct Coche{
    float precio;
    char modelo[100];
    int anyo_matriculacion;
};
```

```
...
```

```
struct Coche *c;
```

```
...
```

```
for (i=0;i<num_coches;i++){
    printf("Introduzca el modelo del coche %d", i+1);
    fgets((c+i)->modelo, 100, stdin);
    printf("Introduzca el precio del coche %d", i+1);
    scanf("%f", &(c+i)->precio);
    printf("Introduzca el año de matriculación del coche %d", i+1);
    scanf("%d", &(c+i)->anyo_matriculacion);
}
```

5. **(34 puntos)** Escribe un programa en lenguaje ensamblador del MIPS que lea una cadena y un entero y llame a una función copia que copie desde la posición indicada por el entero hasta el final la cadena leída en una cadena destino. La función copia recibirá 2 punteros (uno a la cadena original y otro a la cadena destino y un entero) y no devuelve nada. La función main imprimirá la cadena resultante. Ejemplo de funcionamiento: Si leo "Cadena de prueba" y el número 2, se copiará "dena de prueba".

```
.data
cadleida: .space 100
cadcopiada: .space 100
```

```
.globl main
.text
main:
    li $v0, 8
    la $a0, cadleida
    li $a1, 100
    syscall

    li $v0, 5
    syscall
    move $a1, $v0

    la $a0, cadleida
    la $a2, cadcopiada
    jal copia

    li $v0, 4
    la $a0, cadcopiada
    syscall

    li $v0, 10
    syscall

copia:
    add $a0, $a0, $a1
bucle:
    lb $t0, ($a0)
    addi $a0, $a0, 1
    sb $t0, ($a2)
    addi $a2, $a2, 1
    bnez $t0, bucle
    jr $ra
```