

## EXAMEN DE SISTEMAS INFORMÁTICOS INDUSTRIALES SOLUCIÓN (PRÁCTICAS)

La duración del examen es de 1 hora y 30 minutos.

JUNIO 2017

En este examen se le pide al alumno realizar dos ejercicios, uno en C y otro en C++. Para facilitar el inicio, se proporcionan tres ficheros, examen.cpp, pila.h y pila.cpp. Deberá crearse un proyecto en C++, añadir los 3 ficheros y compilarlo.

- En la primera parte se da la estructura para realizar el programa en C. El resto del código está comentado.
- Para realizar el programa en C++ deberá comentarse la primera parte del código y descomentar la siguiente parte. En la parte de programación en C++ se utilizará también la clase pila cuyo prototipo se encuentra en el fichero pila.h y los métodos ya realizados así como los que deben realizarse se encuentran en el fichero pila.cpp.

1. Debe realizarse un programa en C que a partir de una cadena de caracteres, almacene cada palabra en una fila de una matriz. A continuación cambiará cada palabra de un determinado número de caracteres por otra palabra y volverá a generar una cadena de caracteres. Los pasos a seguir son los siguientes:

- 1.1. Contar el número de palabras de la **cadena** y mostrarlo por pantalla. (1.00 puntos)
- 1.2. Realizar una reserva dinámica de memoria para una matriz de tantas filas como palabras y **10** columnas. (0.75 puntos)
- 1.3. Copiar cada palabra en una fila de la matriz (no olvidar cerrar cada cadena de caracteres). Si no se es capaz de completar hasta este punto puede continuarse con el punto 4 utilizando la matriz **m** ya preparada. (1.00 puntos)
- 1.4. Mostrar cada palabra almacenada en la matriz. (0.50 puntos)
- 1.5. Sustituir cada palabra que tenga una longitud igual al valor contenido en **n\_letras** por la palabra “muchas”. (1.00 puntos)
- 1.6. Mostrar de nuevo cada palabra almacenada en la matriz. (0.50 puntos)
- 1.7. Copiar la nueva lista de palabras a la variable **cadena2** separadas por espacios (1.00 puntos)
- 1.8. Mostrar la **cadena2** modificada. (0.25 puntos)

A continuación se muestra un ejemplo de lo que debería salir por pantalla:

```
Numero de palabras: 6
Fila 0: Esto
Fila 1: es
Fila 2: una
Fila 3: frase
Fila 4: de
Fila 5: prueba

Fila 0: Esto
Fila 1: muchas
Fila 2: una
Fila 3: frase
Fila 4: muchas
Fila 5: prueba
Cadena completa modificada: Esto muchas una frase muchas prueba
```



SOLUCIÓN:

```
char cadena[]="Esto es una frase de prueba";
// Matriz auxiliar para utilizar a partir del punto 3 si no sabe crearse a
partir de cadena
char
m[6][10]={{'E','s','t','o','\0'},{'e','s','\0'},{'u','n','a','\0'},{'f','r','a','
's','e','\0'},{'d','e'},{'p','r','u','e','b','a','\0'}};
// Otras variables
char cadena2[100];
char **matriz;
int i,j,k,contador=0, n_letras=2;

// 1 Contar el número de palabras y mostrarlo por pantalla
for (i=0;i<strlen(cadena);i++){
    if (cadena[i]==' '){
        contador++;
    }
}
contador++;
printf("Numero de palabras: %d\n",contador);

// 2 Reserva dinámica de memoria de la matriz
// (tantas filas como palabras, y 10 columnas)
matriz = (char **)malloc(contador*sizeof(char*));
for(i=0; i < contador; i++)
    matriz[i] = (char *)malloc(10*sizeof(char));

// 3 Copiar cada palabra en una fila de la matriz.
// Recordar cerrar la cadena
// Si no se es capaz de completar este punto puede
// continuarse en el punto 4 utilizando la matriz auxiliar m ya preparada
for (i=0,k=0;i<contador;i++){
    for (j=0;k<strlen(cadena);j++,k++){
        if (cadena[k]==' '){
            break;
        }
        matriz[i][j]=cadena[k];
    }
    matriz[i][j]='\0';
    k++;
}

// 4 Mostrar cada palabra almacenada en la matriz
for (i=0;i<contador;i++){
    printf("Fila %d: %s\n",i,matriz[i]);
}

// 5 Sustituir cada palabra que tenga una longitud de n_letras
// por la palabra muchas
for (i=0;i<contador;i++){
    if (strlen(matriz[i])==n_letras){
        strcpy(matriz[i],"muchas");
    }
}
cout << endl;

// 6 Mostrar de nuevo cada palabra almacenada en la matriz
for (i=0;i<contador;i++){
    printf("Fila %d: %s\n",i,matriz[i]);
}
```



# Escuela Politécnica Superior de Elche

---

*Grado en Ingeniería Electrónica y Automática Industrial*

```
// 7 Copiar la lista nueva de palabras en la variable cadena2
// separadas por espacios
strcpy(cadena2,matriz[0]);
for (i=1;i<contador;i++){
    if (i!=contador){
        strcat(cadena2," ");
    }
    strcat(cadena2,matriz[i]);
}

// 8 Mostrar la cadena2 modificada
printf("Cadena completa modificada: %s\n",cadena2);
```

2. En este ejercicio a realizar en C++ se pide realizar un método de la clase pila y una sobrecarga del operador =. En el fichero pila.cpp ya se encuentran los prototipos de dichas funciones vacías para que el alumno sólo deba rellenar ambos métodos. La clase pila es la estudiada en clase y ya tiene programados los constructores, el destructor, y varios métodos para sacar y meter datos en la pila así como un nuevo método mostrar que permite visualizar el contenido de la pila. Los pasos a seguir son los siguientes:

- 1.1. En primer lugar se pide sobrecargar el operador = para que cuando se igualen dos objetos de la clase pila, al que está a la izquierda del igual se le saquen tantos elementos como la dimensión del objeto a la derecha del igual. En el código proporcionado se ha creado ya un objeto pila1 y se han introducido los caracteres de una cadena. A continuación se ha creado un objeto pila2 de 5 elementos. Al igualar pila1=pila2;, pila2 no se verá modificado y pila1 deberá tener 5 elementos menos.

(2.00 puntos)

### Salida del código

```
Mostrar pilal
Esto es una frase de prueba

Mostrar pilal tras sacar tantos caracteres como dimension tiene pila2
Esto es una frase de p
Presione una tecla para continuar . . .
```

- 1.2. En el segundo código se pide crear el método Sacar\_mod. El método Sacar se encargaba de extraer el último elemento de la pila. En el nuevo método se sacará el elemento indicado como parámetro debiendo mover el resto de elementos hacia la izquierda para rellenar el hueco. Si el elemento que pide sacarse es mayor a la cantidad de información introducida en la pila se indicará por pantalla. En el código proporcionado puede verse el contenido de la cadena antes de extraer ningún elemento. A continuación se pide extraer el elemento 5 y vuelve a mostrarse el contenido de la cadena.

(2.00 puntos)

### Salida del código

```
Mostrar pilal
Esto es una frase de prueba

Mostrar pilal antes de quitar nada
Esto es una frase de prueba

Caracter sacado: e

Mostrar pila3 tras sacar el caracter 5
Esto s una frase de prueba
Presione una tecla para continuar . . .
```



SOLUCIÓN:

```
pila& pila::operator = (const pila & pila_orig)
{
    // Código a completar por el alumno
    // Sacará de la pila a la izquierda del igual tantos
    // caracteres como dimensión tenga la pila a la derecha del igual
    // Si la pila a la izquierda del igual se queda sin caracteres
    // dejará de sacar
    for (int i=0;i<pila_orig.tam;i++){
        if (top >=1){
            Sacar();
        }
    }
    // -----
    return (*this);
}

char pila::Sacar_mod(int n)
{
    // Código a completar por el alumno
    // Saca el caracter de la posición n y mueve el resto de
    // caracteres para ocupar su hueco
    // Si la posición n es mayor de las posiciones ocupadas
    // en la pila se indicará por pantalla
    char temp;
    if (n>top){
        cout << "El valor solicitado para sacar no existe" << endl;
    }
    else{
        temp=ppila[n];
        for (int i=n;i<top-1;i++){
            ppila[i]=ppila[i+1];
        }
        top--;
    }

    return(temp);
    // -----
}
```



CÓDIGOS PROPORCIONADOS:

## Examen.cpp

---

```
#include <iostream>
#include <cstdlib>
#include <stdio.h>
#include <locale.h>
#include <string.h>

#include "pila.h"

using namespace std;

int main()
{
    // EJERCICIO 1: PARTE DE C

    char cadena[]="Esto es una frase de prueba";
    // Matriz auxiliar para utilizar a partir del punto 3 si no sabe crearse a
partir de cadena
    char
m[6][10]={{'E','s','t','o','\0'},{'e','s','\0'},{'u','n','a','\0'},{'f','r','a','
's','e','\0'},{'d','e'},{'p','r','u','e','b','a','\0'}};
    // Otras variables
    char cadena2[100];
    char **matriz;
    int i,j,k,contador=0, n_letras=2;

    // 1 Contar el número de palabras y mostrarlo por pantalla

    // 2 Reserva dinámica de memoria de la matriz
// (tantas filas como palabras, y 10 columnas)

    // 3 Copiar cada palabra en una fila de la matriz.
// Recordar cerrar la cadena
// Si no se es capaz de completar este punto puede
// continuarse en el punto 4 utilizando la matriz auxiliar m ya preparada

    // 4 Mostrar cada palabra almacenada en la matriz

    // 5 Sustituir cada palabra que tenga una longitud de n_letras
// por la palabra muchas

    // 6 Mostrar de nuevo cada palabra almacenada en la matriz

    // 7 Copiar la lista nueva de palabras en la variable cadena2
// separadas por espacios

    // 8 Mostrar la cadena2 modificada

    system("PAUSE");
    return 0;

    // EJERCICIO 2.1: PARTE DE C++

    /*
    char cadena[]="Esto es una frase de prueba";
    int i;
    pila pilal;    // Declara una pila de 200 elementos

    // Introduce los elementos de la cadena en la pila uno a uno
for (i=0;i<strlen(cadena);i++){
```



# Escuela Politécnica Superior de Elche

*Grado en Ingeniería Electrónica y Automática Industrial*

```
pilal.Meter(cadena[i]);
```

```
}
```

```
// Muestra el contenido de la pila  
cout << endl << "Mostrar pilal" << endl;  
pilal.Mostrar();
```

```
pila pila2(5); // Creo una pila de tamaño 5
```

```
// El ejecutar esta línea la sobrecarga del operador igual  
// deberá sacar tantos elementos de la pilal como tamaño  
// tenga la pila2. Si la pila2 tiene un tamaño mayor al de la pilal  
// se sacarán todos los elementos  
pilal=pila2; // Actualmente no hace nada al no estar implementado
```

```
cout << endl << "Mostrar pilal tras sacar tantos caracteres como dimension  
tiene pila2" << endl;  
pilal.Mostrar(); // Deberían haber desaparecido los últimos 5 elementos
```

```
system("PAUSE");  
return 0;
```

```
*/
```

```
// EJERCICIO 2.2: PARTE DE C++
```

```
/*
```

```
char cadena[]="Esto es una frase de prueba";  
int i;  
pila pilal; // Declara una pila de 200 elementos
```

```
// Introduce los elementos de la cadena en la pila uno a uno  
for (i=0;i<strlen(cadena);i++){  
    pilal.Meter(cadena[i]);  
}
```

```
// Muestra el contenido de la pila  
cout << endl << "Mostrar pilal" << endl;  
pilal.Mostrar();
```

```
cout << endl << "Mostrar pilal antes de quitar nada" << endl;  
pilal.Mostrar();
```

```
// Actualmente no sacaré ningun caracter ya que el método aun no está  
implementado
```

```
cout << endl << "Caracter sacado: " << pilal.Sacar_mod(5) << endl;  
cout << endl << "Mostrar pila3 tras sacar el caracter 5" << endl;  
pilal.Mostrar();
```

```
system("PAUSE");  
return 0;
```

```
*/
```

```
}
```

## Pila.cpp

---

```
#include <iostream>
#include <cstdlib>
#include <stdio.h>
#include <locale.h>
#include <string.h>

#include "pila.h"

using namespace std;

pila::pila(const int t)
{
    ppila= new char[t];
    tam=t;
    top=0;
}

pila::pila(const pila &pila_orig)
{
    int i;

    tam = pila_orig.tam;
    top = pila_orig.top;

    ppila= new char[tam];

    for(i=0; i< top; i++)
        ppila[i] = pila_orig.ppila[i];
}

pila::~pila(void)
{
    delete []ppila;
}

void pila::Meter( const char c)
{
    ppila[top] = c;
    top++;
}

char pila::Sacar(void)
{
    --top;    // decrementa en uno la pila
    return ppila[top];
}

int pila::Vacía(void)
{
    if(top==0)
        return 1;
    else
        return 0;
}

int pila::Llena(void)
{
    if(top==tam)
        return 1;
    else
```





# Escuela Politécnica Superior de Elche

*Grado en Ingeniería Electrónica y Automática Industrial*

```
return 0;
}

void pila::Mostrar(void)
{
    for (int i=0;i<top;i++){
        cout << ppila[i];
    }
    cout << endl;
}

pila& pila::operator = (const pila & pila_orig)
{
    // Código a completar por el alumno
    // Sacará de la pila a la izquierda del igual tantos
    // caracteres como dimensión tenga la pila a la derecha del igual
    // Si la pila a la izquierda del igual se queda sin caracteres
    // dejará de sacar

    // -----
}

char pila::Sacar_mod(int n)
{
    // Código a completar por el alumno
    // Saca el caracter de la posición n y mueve el resto de
    // caracteres para ocupar su hueco
    // Si la posición n es mayor de las posiciones ocupadas
    // en la pila se indicará por pantalla
}

```

## Pila.h

---

```
#define LIMITE 100

class pila {

private:
    int tam;          // tamaño de la pila
    char *ppila;     // puntero a la pila
    int top;         // ultima posicion ocupada +1

public:
    pila(const int t=LIMITE);          // constructor
    pila(const pila &pila_orig);      // constructor copia
    ~pila(void);                       // destructor

    void Meter( const char c);        // introduce un caracter en la pila
    char Sacar(void);                 // extrae un caracter de la pila
    int Vacia(void);                  // Comprueba si la pila esta vacia
    int Llena(void);                  // Comprueba si la pila esta llena
    void Mostrar(void);               // Muestra el contenido de la pila

    pila& operator = (const pila & pila_orig);    // operador asignación
    char Sacar_mod(int n);

};

```