

PRACTICA-8-RESUELTA-COMPLETA.pdf



user_2716437



Fundamentos de la Programación



1º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**

**Encuentra el trabajo
de tus sueños**

Participa en retos y competiciones de programación

Ten contacto de calidad con empresas líderes en el sector tecnológico mientras vives una experiencia divertida y enriquecedora durante el proceso.

Únete ahora



Escanéame y
obtén más info!!

NUWE



Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación



Escanéame y
obtén más info!!



UNIVERSIDAD DE MÁLAGA
DPTO. DE LENGUAJES Y CC. DE LA COMPUTACIÓN
E.T.S. DE INGENIERÍA INFORMÁTICA

FUNDAMENTOS DE LA PROGRAMACIÓN
E.T.S.I. Informática. Curso 1º

Práctica Nº 8 Strings

Ejercicios de clase.

1. Diseña un algoritmo que permita traducir una cadena numérica al número natural que representa (podemos suponer que la entrada es correcta). Por ejemplo:

Entrada: 7805 (hay que leer el dato como un string)

Salida: 7805 (hay que mostrar el dato como un int)

2. Una palabra w es un anagrama de la palabra v , si podemos obtener w cambiando el orden de las letras de v . Por tanto, la longitud de ambas palabras es la misma, tienen las mismas letras y el número de repeticiones de cada letra es el mismo en ambas palabras. Por ejemplo, VACA es un anagrama de CAVA. Sin embargo, VACV no es un anagrama de CAVA.

Diseña un algoritmo que lea un texto y determine de cuántas palabras es anagrama la primera que aparece dentro de dicho texto. Hay que tener en cuenta que:

- El texto contiene un número indefinido de palabras.
- El texto termina con la palabra FIN.
- Cada palabra tiene un número indefinido pero limitado de caracteres (letras mayúsculas).
- Las palabras del texto estarán separadas, indistintamente, por espacios en blanco y/o por saltos de línea.

Ejemplo de ejecución (las palabras anagramas de VACA son CAVA y AACV):

Introduzca un texto terminado con la palabra FIN:

VACA NO ES CAVA NI TAMPOCO VACV PERO CASI AACV FIN

El numero de palabras anagramas de la primera es: 2

Ejercicios de refuerzo.

3. Supongamos que deseamos evaluar a un determinado número de alumnos siguiendo el criterio de que aprobará una determinada evaluación aquel que supere o iguale la nota media de la clase en dicha evaluación.

WUOLAH

Diseña un algoritmo que lea por teclado un número de alumnos, N_{alumnos} (como máximo podrá tomar el valor de $\text{MAX_ALUMNOS}=20$), y las notas de tres evaluaciones para cada alumno, $N_{\text{EVALUACIONES}}=3$, y como resultado emita un informe indicando para cada alumno el resultado de cada evaluación (Aprobado o Suspenso).

Ejemplo de la entrada de datos.

```
Introduce el numero de alumnos: 3
Introduce el nombre y 3 notas: Juan  5.3 3.2 8.5
Introduce el nombre y 3 notas: Luis  2.3 5.2 7.5
Introduce el nombre y 3 notas: Pedro 6.3 5.2 6.5
```

Ejemplo de la salida que se debe obtener.

Alumno	Nota-1	Nota-2	Nota-3
Juan	Aprobado	Suspenso	Aprobado
Luis	Suspenso	Aprobado	Aprobado
Pedro	Aprobado	Aprobado	Suspenso

4. Diseña un algoritmo que se comporte como una calculadora que pida repetidamente un operador de conjuntos y dos operandos que sean conjuntos de letras minúsculas y que escriba el resultado de la operación. Las operaciones se expresan como caracteres, siendo válidas las siguientes:

```
+   Unión de conjuntos
-   Diferencia de conjuntos
*   Intersección de conjuntos
```

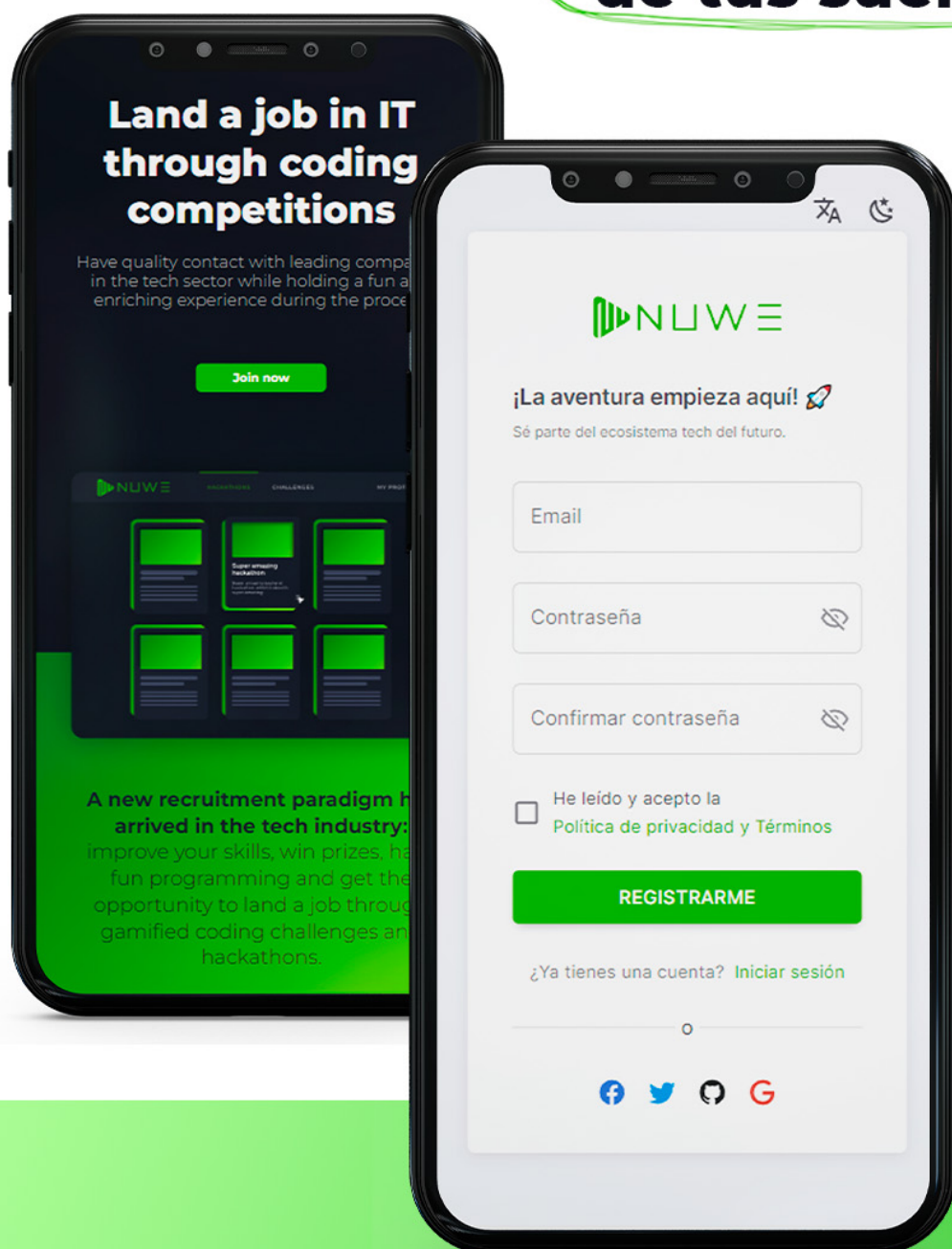
El proceso se repetirá hasta que se introduzca como código de operación el carácter '&'. Los operandos y el resultado se expresan como cadenas de caracteres. Ejemplo:

```
Operación = *
Operando1 = azufre
Operando2 = zafio
Resultado = afz
Operación = -
Operando1 = abril
Operando2 = arco
Resultado = bil
Operación = &
FIN
```

5. Realiza los ejercicios 22 y 23 de la Tercera relación de problemas.



Encuentra el trabajo de tus sueños



Participa en retos y competiciones de programación

Ten contacto de calidad con empresas líderes en el sector tecnológico mientras vives una experiencia divertida y enriquecedora durante el proceso.

Únete ahora

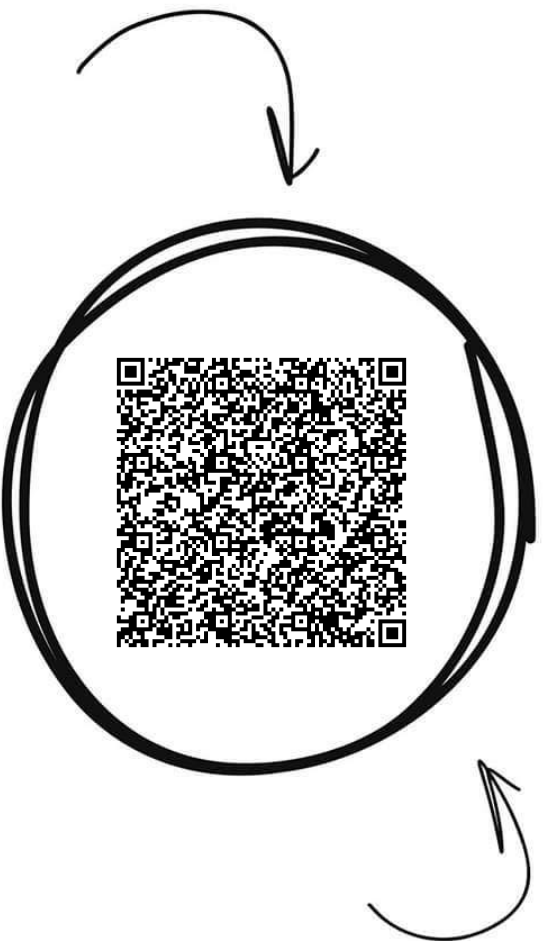
Escanéame y
obten más info!!



Fundamentos de la Programación



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Note bank of the

WUOLAH

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



```

/*
Diseña un algoritmo que permita traducir una cadena numérica al número
natural que
representa (podemos suponer que la entrada es correcta). Por ejemplo:
Entrada: 7805 (hay que leer el dato como un string)
Salida: 7805 (hay que mostrar el dato como un int)
*/

#include <iostream>
using namespace std;

int conv(string c);

int main(){
    string cadena;
    cout<<"Entrada: ";
    cin>>cadena;
    cout<<"Salida: "<<conv(cadena);
}

int conv(string c){
    int res = 0;
    for(int i=0; i<int(c.size()); i++){
        res = res*10 + c[i] - '0';
    }
    return res;
}

```

```

/*
Una palabra w es un anagrama de la palabra v, si podemos obtener w
cambiando el orden
de las letras de v. Por tanto, la longitud de ambas palabras es la misma,
tienen las mismas
letras y el número de repeticiones de cada letra es el mismo en ambas
palabras. Por ejemplo,
VACA es un anagrama de CAVA. Sin embargo, VACV no es un anagrama de CAVA.
Diseña un algoritmo que lea un texto y determine de cuántas palabras es
anagrama la
primera que aparece dentro de dicho texto. Hay que tener en cuenta que:
• El texto contiene un número indefinido de palabras.
• El texto termina con la palabra FIN.
• Cada palabra tiene un número indefinido pero limitado de caracteres
(letras
mayúsculas).
• Las palabras del texto estarán separadas, indistintamente, por espacios
en blanco
y/o por saltos de línea.
Ejemplo de ejecución (las palabras anagramas de VACA son CAVA y AACV):
Introduzca un texto terminado con la palabra FIN:
VACA NO ES CAVA NI TAMPOCO VACV PERO CASI AACV FIN
El numero de palabras anagramas de la primera es: 2
*/

```

```

#include <iostream>
using namespace std;

bool mismasLetras(const string& palabra, const string& posAna);
int buscarLetra(const string& p, char c);
bool anagrama(const string& pal, const string& posAna);

int main(){
    string pal1;
    string palSig;
    int ana = 0;
    cout<<"Introduzca un texto terminado con la palabra FIN: "<<endl;
    cin>>pal1;
    if(pal1 != "FIN"){
        cin>>palSig;
        while(palSig != "FIN"){
            if(anagrama(pal1,palSig)){
                ana++;
            }
            cin>>palSig;
        }
    }
    cout<<"El numero de palabras anagramas de la primera es: "<<ana;
}

bool mismasLetras(const string& palabra, const string& posAna){
    bool ok = true;
    string aux = posAna;
    int i = 0;

```


Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación



Escanéame y
obtén más info!!



```
int j;

while(ok && i<int(palabra.size())){
    j = buscarLetra(aux,palabra[i]);
    if(j<int(aux.size())){
        aux[j] = ' ';
        i++;
    }else{
        ok = false;
    }
}
return ok;
}

int buscarLetra(const string& p, char c){
    int i = 0;
    while(i<int(p.size()) && c!=p[i]){
        i++;
    }
    return i;
}

bool anagrama(const string& pal, const string& posAna){
    bool ok;
    if(pal.size() == posAna.size()){
        ok = mismasLetras(pal,posAna);
    }else{
        ok = false;
    }
    return ok;
}
```



WUOLAH


```

/*
Supongamos que deseamos evaluar a un determinado número de alumnos
siguiendo el
criterio de que aprobará una determinada evaluación aquel que supere o
iguale la nota media
de la clase en dicha evaluación.
Diseña un algoritmo que lea por teclado un número de alumnos, N_alumnos
(como máximo
podrá tomar el valor de MAX_ALUMNOS=20), y las notas de tres evaluaciones
para cada
alumno, N_EVALUACIONES=3, y como resultado emita un informe indicando
para cada
alumno el resultado de cada evaluación (Aprobado o Suspenso).

```

Ejemplo de la entrada de datos.

Introduce el numero de alumnos: 3

Introduce el nombre y 3 notas: Juan 5.3 3.2 8.5

Introduce el nombre y 3 notas: Luis 2.3 5.2 7.5

Introduce el nombre y 3 notas: Pedro 6.3 5.2 6.5

Ejemplo de la salida que se debe obtener.

Alumno Nota-1 Nota-2 Nota-3

```

-----
Juan Aprobado Suspenso Aprobado
Luis Suspenso Aprobado Aprobado
Pedro Aprobado Aprobado Suspenso
*/

```

```

#include <iostream>
#include <array>
using namespace std;

const int N_EVALUACIONES=3;
typedef array<double,N_EVALUACIONES> TNotas;

struct TRegistro{
    string nombre;
    TNotas notas;
};

const int MAX_ALUMNOS=20;
typedef array<TRegistro,MAX_ALUMNOS> TAlumnos;

struct TClase{
    TAlumnos alumnos;
    int indAlum;
};

void leerAlumnos(TClase& clase);
void salida(const TClase& clase);
void notas(const TClase& clase);
int media(const TClase& clase, int j);

int main(){

```

```

    TClase clase;
    cout<<"Introduce el numero de alumnos: ";
    cin>>clase.indAlum;
    leerAlumnos(clase);
    salida(clase);
}

void leerAlumnos(TClase& clase){
    for(int i=0; i<clase.indAlum; i++){
        cout<<"Introduce el nombre y "<<N_EVALUACIONES<<" notas: ";
        cin>>clase.alumnos[i].nombre;
        for(int j=0; j<N_EVALUACIONES; j++){
            cin>>clase.alumnos[i].notas[j];
        }
    }
}

void salida(const TClase& clase){
    cout<<"Alumno ";
    for(int i=0; i<N_EVALUACIONES; i++){
        cout<<"Nota-"<<i<<" ";
    }
    cout<<endl;
    cout<<"-----"<<endl;
    notas(clase);
}

void notas(const TClase& clase){
    for(int i=0; i<clase.indAlum; i++){
        cout<<clase.alumnos[i].nombre<<" ";
        for(int j=0; j<N_EVALUACIONES; j++){
            if(clase.alumnos[i].notas[j]<media(clase, j)){
                cout<<"Supenso ";
            }else{
                cout<<"Aprobado ";
            }
        }
        cout<<endl;
    }
}

int media(const TClase& clase, int j){
    double sol = 0;
    double cont = 0;
    for(int i=0; i<clase.indAlum; i++){
        sol+=clase.alumnos[i].notas[j];
        cont++;
    }
    return sol/cont;
}

```

```

/*
Diseña un algoritmo que se comporte como una calculadora que pida
repetidamente un
operador de conjuntos y dos operandos que sean conjuntos de letras
minúsculas y que
escriba el resultado de la operación. Las operaciones se expresan como
caracteres, siendo
válidas las siguientes:
+ Unión de conjuntos
- Diferencia de conjuntos
* Intersección de conjuntos

El proceso se repetirá hasta que se introduzca como código de operación
el carácter '&'. Los
operandos y el resultado se expresan como cadenas de caracteres. Ejemplo:
Operación = *
Operando1 = azufre
Operando2 = zafio
Resultado = afz
Operación = -
Operando1 = abril
Operando2 = arco
Resultado = bil
Operación = &
FIN
*/

#include <iostream>
using namespace std;

bool esta(char c, const string& pal);
void opUnion(const string& op1, const string& op2, string& res);
void opDif(const string& op1, const string& op2, string& res);
void opInter(const string& op1, const string& op2, string& res);

int main(){
    char operacion;
    string op1,op2,res;
    cout<<"Operación = ";
    cin>>operacion;
    while(operacion!='&'){
        if(operacion!='+' && operacion!='-' && operacion!='*'){
            cout<<"ERROR; operacion no valida"<<endl;
        }else{
            cout<<"Operando1 = ";
            cin>>op1;
            cout<<"Operando2 = ";
            cin>>op2;
            switch (operacion){
                case '+' : opUnion(op1,op2,res);
                    break;
                case '-' : opDif(op1,op2,res);
                    break;
                case '*' : opInter(op1,op2,res);
            }
        }
    }
}

```

Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación



Escanéame y
obtén más info!!



```
                break;
            }
            cout<<"Resultado = "<<res<<endl;
        }
        cout<<"Operación = ";
        cin>>operacion;
    }
    cout<<endl<<"FIN";
}

bool esta(char c, const string& pal){
    int i=0;
    while(i<int(pal.size()) && c!=pal[i]){
        i++;
    }
    return i<int(pal.size());
}

void opUnion(const string& op1, const string& op2, string& res){
    res = op1;
    for(int i=0; i<int(op2.size()); i++){
        if(!esta(op2[i],op1)){
            res+=op2[i];
        }
    }
}

void opDif(const string& op1, const string& op2, string& res){
    res ="";
    for(int i=0; i<int(op1.size()); i++){
        if(!esta(op1[i],op2)){
            res+=op1[i];
        }
    }
}

void opInter(const string& op1, const string& op2, string& res){
    res ="";
    for(int i=0; i<int(op1.size()); i++){
        if(esta(op1[i],op2)){
            res+=op1[i];
        }
    }
}
```



WUOLAH

