Ejemplos TP3

Uso del TDA **string, vector, array** y **Templates en funciones**

```
#include <iostream>
//#include <string>
#include <vector>
#include <array>
using namespace std;


//Uso de template en funciones
template <class T>
T suma( T a, T b){
    return a+b;
}

/*
//funciones reemplazadas por el template
int suma(int a, int b){
    return a+b;
}

float suma (float a, float b){
    return a+b;
}

string suma(string a, string b){
    return a+b;
}*/

int main(int, char **)
{

 /*
    //Probando la clase vector que contiene string
    cout << "Hello, world!\n";
    vector<string> misPalabras;
    cout<<misPalabras.size()<<endl;
    misPalabras.push_back("hola");
    misPalabras.push_back("Adios");
    cout<<misPalabras.size()<<endl;

    vector<string> a(10,"Hace calor");
    cout<<endl;
    for(string s:misPalabras)
       cout<<s<<", ";
    cout<<endl;

    for(string s:a)
        cout<<s<<", ";

    misPalabras.assign(a.begin(),a.end());
    cout<<endl;
    for(string s:misPalabras)
            cout<<s<<", ";

    //misPalabras.insert(misPalabras.begin()+1,a.begin(),a.end()-9);

    misPalabras.insert(misPalabras.end()-1,"andaaaa");

    cout<<endl;
    for(string s:misPalabras)
       cout<<s<<", ";

    misPalabras.erase(misPalabras.begin()+1);
    cout<<endl;
    for(string s:misPalabras)
          cout<<s<<", ";
    misPalabras.front()="Buenas tardes ";
    cout<<endl<<misPalabras.front();
*/

 /*
    //Probando la clase string
    string s="0";
    string s1 = s;
    s1.append("12345");
    //s1.append(14);
    s.append(s1,2,3);
    cout<<s<<" "<<s1<<endl;

    for(size_t i=0;i<s1.size();i++)
        cout<<s1.at(i)<<", "; //cout<<s1[i]<<", ";

    cout<<endl;

    for(string::iterator i=s1.begin();i!=s1.end();i++)
         cout<<*i<<", ";
    cout<<endl;

    for(string::const_reverse_iterator i=s1.crbegin();i!=s1.crend();i++){
         //cout<<*i<<", ";
         //*i='a';
         cout<<*i<<", ";
    }
    cout<<endl;

    for(char a: s1)
        cout<<a;
    */
    /*
        //Probando el template
        int i=10, ii=20;
        int iii=suma(i,ii);
        cout<<iii<<endl;
        float a=10.00, b=12.44444;
        float c=suma(a,b);
        cout<<c<<endl;
        string dd="hola", ee="aca estoy";
        string ff=suma(dd,ee);
        cout<<ff; */

        //Probando la clase array
        array<int, 5> a ={10,11,12,13,14};

        for(int i:a)
            cout<<i<<", ";

        cin>>(a.at(5)); //cin>>(a[5]));
        for(int i:a)
            cout<<i<<", ";


    return 0;

}
```

Uso del TDA **string**

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

int main()
{
    //Constructores
    string s1;
    s1 = "Hola!!";
    string s2 = "Estamos en EDyA.";
    string s3("https://campusvirtual.ing.unlpam.edu.ar/course/view.php?id=176&section=0");
    string s4(1, char(32));
    string s5(3, 'w');
    string s6 (s3, 8, 6);
    string s7 ("Chau!! Hasta otro dia");
    cout << "\ns1: " << s1 << "\ns2: "  << s2 << "\ns3: " << s3 << "\ns4: " << s4 << "\ns5:

    string msg = s1+s2+s3+s5+s2+s4+s7+s4+s6;
    cout << "\nmsg: " << msg;

    //Sobrecarga de operadores
    s2 = "Estamos en la catedra de EDyA";
    cout << "\ns2 despues de la asignacion: " << s2;
    s7+= " a las 15:00."; //concatenaci�n
    cout << "\ns7 despues de la concatenacion: " << s7;
    cout << "\ncompara s4==s2: " << (s4==s2);
    cout << "\ncompara (s1+s2+s3+s5+s2+s4+s7+s4+s6)==msg: " << ((s1+s2+s3+s5+s2+s4+s7+s4+s6)
    cout << "\ncompara (s1+s2+s3+s5+s2+s4+s7+s4+s6)==(s1+s2+s3+s5+s2+s4+s7+s4+s6): " << ((s1
    cout << "\ncompara msg!=msg: " << (msg!=msg);
    cout << "\n msg[0] y msg[15]:" << msg[0] << " " << msg[15];

    //Algunos m�todos �tiles
    string a ="Estructuras es muy ";
    cout << "\na " << a;
    a.insert( a.size(), "ALGO!!"); //igual a a.length()
    cout << "\na: " << a;
    cout << "\n msg[0] y msg[15]: " << msg[0] << " " << msg[15];
    a.erase(0,12);
    cout << "\na: " << a;
    a= "Estructuras es muy Algo";
    string b = a.substr(19);
    cout << "\n b y a: " << b << " " << a;

    return 0;
}
```

```
string metodo1(){
    string s1(3, char(119));
    string s2="ing.unlpam.edu.ar";
    string s3=s1+s2;
    s3.insert( 3, ".");
    cout<<endl<<" Metodo 1 "<<endl<<" s1: "<<s1<<endl<<" s2: "<<s2<<endl<<" s3: "<<s3<<endl;
    return s3;
}

string metodo2(){
    string s1(3, 'w');
    string s2("ing.unlpam.edu.ar");
    string s3 = s1.assign(s1+s2);
    s3.assign(s1.substr(0,3));s3.push_back('.');s3.append(s2);
    cout<<endl<<" Metodo 2 "<<endl<<" s1: "<<s1<<endl<<" s2: "<<s2<<endl<<" s3: "<<s3<<endl<
    return s3;
}

int mainSrtrings()
{
    string s1m = metodo1();
    string s2m = metodo2();
    cout<<"Comparacion metodos: ";
    cout <<(s1m==s2m?"True":"False")<<endl;
    s1m.erase(0, s1m.size());
    s2m.clear();

    cout<<"Comparacion tamanios: ";
    cout <<(s1m.length()==s2m.size()?"True":"False")<<endl<<endl;

    cout<<"Main "<<endl <<" s1m: "<< s1m<<endl<<" s2m: "<<s2m;
    iterador();
    return 0;
}
```

Uso de **iteradores**

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << " Probando iteradores de string" << endl;
    string  palabra = "yo dono rosas oro no doy";
    cout<<" ";
    for ( string::iterator it=palabra.begin(); it!=palabra.end(); ++it)
       cout << (char)(*it-32);

    cout<<  endl << " ";

   string::reverse_iterator rit=palabra.rbegin();
   while(rit!=palabra.rend()){
       cout << *rit;
       ++rit;
   }

    cout<<  endl << " ";

   for ( string::iterator it=palabra.begin(); it!=palabra.end(); ++it){
        *it=148;
         cout << *it;
   }
    cout<<  endl << " ";

   for (auto itc=palabra.cbegin(); itc!=palabra.cend(); itc++){
    std::cout << *itc;
   }

   cout<<  endl;
   return 0;
}
```

Uso del TDA **vector**

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    vector<int> miVector;
    cout<<" Cantidad de elementos "<<miVector.size();
    vector<int>::iterator it = miVector.begin();
    for(int i=0;i<5;i++){
         it = miVector.begin();
         miVector.insert ( it , i );
    }
    cout<<endl<<" ";
    for(auto i=miVector.rbegin();i!=miVector.rend();i++){
            cout << *i <<" ";
    }
    cout<<endl<<" ";
    for(auto i=miVector.begin();i!=miVector.end();i++){
            cout << *i <<" ";
    }
    cout<<endl<<" Cantidad de elementos "<<miVector.size();
    cout<<endl<<" Cantidad maxima de elementos soportados "<<miVector.max_size()<<endl;

    int suma=0;
    while (!miVector.empty()) {
     suma += miVector.front();
     miVector.pop_back();
    }
    cout<<endl<<" Suma "<<suma<<endl;
    cout<<" Cantidad de elementos "<<miVector.size();

    return 0;
}
```