

		PRIMER PARCIAL SOLUCIONARIO		CÓDIGO DEL ESTUDIANTE	
CARRERA: INGENIERÍA DE SISTEMAS			ASIGNATURA: Programación II		FECHA: 20/03/2022
CURSO: Tercer Semestre			DOCENTE: Lic. Marlene Ticona Flores		
UNIDADES TEMÁTICAS A EVALUAR		1. Programación Modular 2. Cadenas			
<p align="center">RECOMENDACIONES A LOS ESTUDIANTES</p> <ol style="list-style-type: none">Los estudiantes tienen 5 (Cinco) minutos para interpretar el examen y solicitar aclaraciones al docente.El RAC-07 (RÉGIMEN DISCIPLINARIO), en el CAP IV. FALTAS Y SANCIONES, Art. 20 tipifica el FRAUDE O INTENTO DE FRAUDE EN EXÁMENES, como “CAUSAL DE SEPARACIÓN SIN DERECHO A REINCORPORACIÓN” de la EMI.Mediante MOODLE el estudiante descargará el examen y subirá el examen resuelto en formato PDFMediante TEAMS el estudiante está en la obligación de permanecer conectado durante el desarrollo de la pruebaTiempo de Duración:<ol style="list-style-type: none">“90 Minutos” para resolver el EXAMENOtras que el docente considere necesarias.					

- La empresa de transporte “TRANS-VELOZ” desea llevar un control sobre el proceso de reserva de los asientos en los buses de transporte. Para esto se preparó un formulario que permite registrar el estado de reserva de los asientos de la siguiente manera:

RESERVA PASAJES BUS CAMA LEITO

Asiento	Estado	Ubicación	Precio	CI
1	R	“Panorámico”	40	“3578958”
2	L	“Panorámico”	40	-
3	L	“Panorámico”	40	-
4	R	“Panorámico”	40	“6859745”
-	-	-	-	-
11	R	“Bajo”	35	“6068749”
20	L	“Bajo”	35	-

La tabla refleja el estado de cada asiento: **(R)** Reservado, **(L)** Libre.
 Los primeros 10 asientos están en Ubicación “Panorámico”.

Analizar y diseñar una solución para resolver:

- Realizar función(es) o procedimiento(s) para mostrar Asientos libres (mostrar número de asiento y ubicación) **(2 punto)**
- Política de la empresa dice que una persona no puede realizar más de una reserva con un mismo número de carnet, es así que dado un número se debe verificar que el carnet no exista. Realizar una función o procedimiento para buscar el número de carnet, si no existe realizar reserva caso contrario enviar mensaje de que el pasajero no puede realizar otra reserva **(2 puntos)**
- Realizar función(es) o procedimiento(s) que se requiera para realizar una nueva reserva de asiento que se tenga uno disponible (Tomar en cuenta que los asientos están numerados ascendentemente) **(2 puntos)**

SOLUCIÓN EJERCICIO 1

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

using namespace std;

/*PROTOTIPO DE FUNCIONES Y PROCEDIMIENTOS*/

void cargar_datos(char estado[],int asiento[],string carnet[21]);
void mostrar_libres(char estado[],int libre[21]);
void mostrar_planilla(char estado[],string carnet[21]);
void reservar(char estado[],string carnet[10]);
int busca_carnet(string carnet[10],char ci[]);

/*PROGRAMA PRINCIPAL*/
int main()
{
    int asiento[21],op;
    char estado[21];
    int libre[21];
    string carnet[21];

    system("color 1e");
    cargar_datos(estado,asiento,carnet);
    do{
        cout<< "-----*TRANS BOLIVIA*-----"<<endl;
        cout<< "-----OPCIONES-----"<< endl;
        cout<< "1. MOSTRAR ASIENTOS LIBRES      "<< endl;
        cout<< "2. RESERVAR ASIENTO              "<< endl;
        cout<< "3. Salir                                "<< endl;
        cout<< "-----\n"<< endl;
        cin >> op;
        switch(op){
            case 1 : mostrar_libres(estado,libre);
                    break;
            case 2 : reservar(estado,carnet);
                    mostrar_planilla(estado,carnet);
                    break;
            case 3 :break;
        }
    }while(op!=4);
    return 0;
}

//PROCEDIMIENTO CARGA DE DATOS AL PROGRAMA
void cargar_datos(char estado[],int asiento[],string carnet[]){
    int i;
    for( i=0;i<20;i++){
        asiento[i]=i+1;
        estado[i]='L';
        carnet[i]=" ";
    }
}

//PROCEDIMIENTO QUE MUESTRA ASIENTOS LIBRE
void mostrar_libres(char estado[],int libre[]){
    int i;
    int j=0;
    cout<<"-----"<<endl;
    cout<<"\nLOS ASIENTOS LIBRES DEL BUS SON:\n";
    cout<<"-----"<<endl;

    for(i=0;i<20;i++){
        if(estado[i]=='L'){
            libre[j]=i+1;
            j++;
        }
    }
}
```

```

        cout<<"ASIENTO\tESTADO\tUBICACION\tCI"<<endl;
        for(int i=0;i<j;i++){
            int asientoReser=libre[i];
            cout<<asientoReser<<"\t"<<estado[asientoReser-1]<<"\t";
                if (i<10 )
                    cout<<"Panoramico"<<endl;
                else
                    cout<<"Bajo"<<endl;
        }
        cout << endl;
    }
//PROCEDIMIENTO MOSTRAR PLANILLA
void mostrar_planilla(char estado[],string carnet[]){

    cout<<"-----"<<endl;
    cout<<"\nLOS ASIENTOS LIBRES Y RESERVADOS DEL BUS SON:\n";
    cout<<"-----"<<endl;
        cout<<"ASIENTO\tESTADO\tUBICACION\tCI"<<endl;
        for(int i=0;i<20;i++){

            cout<<i+1<<"\t"<<estado[i]<<"\t";
            if (i<10 )
                cout<<"Panoramico"<<"\t";
                else
                    cout<<"Bajo"<<"\t";
                cout<<carnet[i]<<endl;
        }
        cout << endl;
    }
//FUNCION QUE BUSCA CARNET
int busca_carnet(string carnet[],char ci[]){
    bool band=false;
    int i,posi;
    for(i=0;i<=20;i++){
        if(carnet[i]== ci){
            band=true;
            posi=i;
        }
    }
    if (band==true)
        return posi;
    else
        return -1;
    }
//PROCEDIMIENTO RESERVA DE PASAJE
void reservar(char estado[],string carnet[]){
    int r,asi;
    char ci[10];
    cout << "Ingrese el CI para la reserva" << endl;
    cin>>ci;
    cout << endl;
    r=busca_carnet(carnet,ci);
    if(r == -1)
    {
        cout<<"Que numero de asiento desea ";
        cin>>asi;
        estado[asi-1]='R';
        carnet[asi-1]=ci;
        cout<<"Felicidades ya tiene reservado su boleto!!!"<<endl;

    }
    else
    {
        cout << "Error de reserva Numero de CI ya fue usado\n" << endl;
    }
}

```

- 2 Una cadena de caracteres capicúa es la que se lee igual de derecha a izquierda o viceversas. Por ejemplo: “**la ruta natural**”.
- a) Escriba una función en c++ que, dada una cadena, determine si es o no capicúa. **(2pts.)**
 - b) Si es capicúa contar las palabras, caso contrario mostrar la cadena invertida. **(2pts.)**
- Incluir librería **Cadenas.h** en el programa, la librería **Cadenas.h** debe contener los subprocesos que permitan contar palabras y mostrar histograma.

PROGRAMA CPP

```
#include <stdio.h>
#include <iostream>
#include <string.h>
#include "CADENAS.h"
using namespace std;
bool EsCapicua(char texto[]);
int main()
{
    char texto[255];
    int vocales[6];
    int t;
    cout<<"Escriba un Frase: ";
    fflush(stdin);
    gets(texto);
    fflush(stdin);
    t=strlen(texto);
    if (EsCapicua(texto))
    {
        cout<<"EL TEXTO ES CAPICUA!!"<<endl;
        cout<<"_____"<<endl;
        cout<<"LA CANTIDAD DE PALABRAS DEL TEXTO ES="
        "<<ContarPalabras(texto)<<endl;
    }
    else
    {
        cout<<"EL TEXTO NO ES CAPICUA!!"<<endl;
        cout<<"_____"<<endl;
        cout<<"LA CADENA INVERTIDA ES:"<<endl;
        cout<<"_____"<<endl;
        Invertir_Cadena(texto,t);
        cout<<texto;
    }
    return 0;
}
bool EsCapicua(char texto[])
{
    int p = 0;
    int q = strlen(texto)-1;
    bool band = true;
    while (p<q) {
        if(texto[p]!=texto[q]){
            if(texto[p]==' '){
                p++;
            }
            else {
                if(texto[q]==' '){
                    q--;
                }
                else{
                    band = false;
                    p++;
                    q--;
                }
            }
        }
    }
}
```

```

        else
        {
            p++;
            q--;
        }
    }
    return band;
}

```

HEADER

```

#ifndef CADENAS_H_INCLUDED
#define CADENAS_H_INCLUDED
#include <string.h>
using namespace std;

/**funcion contar palabras*/
int ContarPalabras(char texto[])
{
    int cont = 0;
    int espacio = 1;
    int i,tam=strlen(texto);
    for(i=0; i < tam; i++)
    {
        if ( (texto[i]==' ') && (i > 0) && (texto[i-1]!=' ') )
        {
            cont++;
            espacio = 1;
        }
        else
            espacio = 0;
    }
    if (espacio==0) cont++;
    return cont;
}

/**procedimiento invertir caena*/
void Invertir_Cadena(char cade[],int t){
    int i,j;
    j=t-1;
    char aux;
    for(i=0;i<t/2;i++){
        aux=cade[i];
        cade[i]=cade[j];
        cade[j]=aux;
        j--;
    }
}

#endif // CADENAS_H_INCLUDED

```