Universidad Nacional Abierta
Vicerrectorado Académico
Área de Ingeniería
Carrera de Ingeniería de Sistemas

Trabajo Práctico Sustitutivo

Asignatura: **Computación II** Código: **324**

Fecha de Devolución: **19/09/2022**

Nombre del Estudiante: **David Alfonso Olivo Rodríguez**

Cédula de Identidad: **29.959.060**

Centro Local / Unidad de Apoyo: **Centro Local Aragua**

Correo Electrónico: davidoar15@gmail.com Teléfono Celular: **0414-0529486**

Carrera: **Ingeniería de Sistemas (Cód. 236)**

Número de Originales: **1**

Lapso: **2022-1**

Resultados de Corrección

	Objetivos			
Logrado: 1	**1**	**2**	**3**	**4**
No Logrado: 0				

Cuerpo del Trabajo

M: 1, U: 1, O: 1

1. Se requiere leer la temperatura todos los mediodías en la ciudad de Cabimas, durante un mes y luego informar la temperatura promedio mensual, así como el día más caluroso y el más frío. Sabiendo que la temperatura mínima de la tierra es de -90° Centígrados y la máxima es de 60° Centígrados, se pide:

Haciendo uso de arreglos, hacer un programa en C++, que:

a. Calcule el promedio mensual de la temperatura.

b. La temperatura máxima y mínima registrada.

Respuesta:

Código en C++:

```
#include<iostream>
#include<stdlib.h>
using namespace std;

int main(){
    float temps[100], mayor = 0, menor = 1000, sumatoria = 0, Promedio = 0;
    int n;

    cout<<"Inserte el numero de dias del Mes: "; cin>>n;

    for(int i=0; i<n; i++){
        cout<<"Dia "<<i+1<<". Inserte la Temperatura, en Grados Centigrados: ";
        cin>>temps[i];
        if(temps[i] > mayor){
            mayor = temps[i];
        }
        if(temps[i] < menor){
            menor = temps[i];
        }
    }
```

```
    for(int x=0;x<n;x++){
        sumatoria += temps[x];
    }

    Promedio = sumatoria/n;

    cout<<"\nEl Promedio Mensual de la Temperatura, en Grados Centigrados, es: "<<Promedio<<endl;
    cout<<"\nLa Maxima Temperatura registrada fue: "<<mayor<<endl;
    cout<<"\nLa Minima Temperatura registrada fue: "<<menor<<endl;

    cout<<"\n";
    system("pause");
    return 0;
}
```

M: 1, U: 2, O: 2

2. Implemente una función C++ que recibe una lista de enteros L y un número entero n de forma que modifique la lista mediante el borrado de todos los elementos de la lista que tengan este valor.

Respuesta:

Código en C++:

```
#include<iostream>
#include<stdlib.h>
#include<list>
using namespace std;

//Estructura de Lista
struct Nodo{
    int dato;
    Nodo *siguiente;
};

//Prototipos de Funciones:
void menu();
void insertarLista(Nodo *&, int);
void mostrarLista(Nodo *);
Nodo* eliminarElemento(Nodo *&, int);

Nodo *Lista = NULL;
int main(){
    int dato;

    menu();

    cout<<"\n";
    system("pause");
    return 0;
}
```

```
//Función Menú para Lista:
void menu(){
    int opcion, dato;

    do{
        cout<<"\t.:MENU:.\n";
        cout<<"1. Insertar elemento a la LISTA\n";
        cout<<"2. Mostrar los elementos de la LISTA\n";
        cout<<"3. Eliminar un nodo de la LISTA\n";
        cout<<"4. Salir\n";
        cout<<"Opcion: "; cin>>opcion;

        switch(opcion){
            case 1: cout<<"Inserte un numero: "; cin>>dato;
                    insertarLista(Lista, dato);
                    cout<<"\n";
                    system("pause");
                    break;

            case 2: mostrarLista(Lista);
                    cout<<"\n";
                    system("pause");
                    break;

            case 3: cout<<"\nInserte el elemento a eliminar: "; cin>>dato;
                    eliminarElemento(Lista, dato);
                    cout<<"\n";
                    system("pause");
                    break;
        }
        system(“cls”);
    }while(opcion != 4);
}
```

```
//Función Insertar:
void insertarLista(Nodo *&Lista, int n){
    Nodo *nuevo_nodo = new Nodo();
    nuevo_nodo->dato = n;

    Nodo *aux1 = Lista;
    Nodo *aux2;

    while((aux1 != NULL) && (aux1->dato < n)){
        aux2 = aux1;
        aux1 = aux1->siguiente;
    }
    if(Lista == aux1){
        Lista = nuevo_nodo;
    }
    else{
        aux2->siguiente = nuevo_nodo;
    }
    nuevo_nodo->siguiente = aux1;
    cout<<"\tElemento "<<n<<" insertado a LISTA correctamente\n";
}

//Función Mostrar Lista:
void mostrarLista(Nodo *Lista){
    Nodo *actual = new Nodo();
    actual = Lista;

    while(actual != NULL){
        cout<<actual->dato<<" -> ";
        actual = actual->siguiente;
    }
}
```

```
//Función Requerida para Eliminar:
Nodo* eliminarElemento(Nodo *&Lista, int n){
    Nodo *aEliminar;
    Nodo *aux = Lista;

    while(aux != NULL){
        if(Lista->dato == n){
            aEliminar = Lista;
            Lista = Lista->siguiente;
            aux = aux->siguiente;
            delete aEliminar;
        }
        else{
            if((aux->siguiente != NULL) && (aux->siguiente->dato == n)){
                aEliminar = aux->siguiente;
                aux->siguiente = aux->siguiente->siguiente;
                delete aEliminar;
            }
            else{
            aux = aux->siguiente;
            }
         }
    }
    return Lista;
}
```

M: 2, U: 3, O: 3

3. Escribir una función Reemplazar en C++ que tenga como argumentos una pila con tipos de elementos int y dos valores int: Nuevo y Viejo; de forma que, si el segundo valor aparece en el lugar de la pila, sea reemplazado por el primero.

Respuesta:

Código en C++:

```
#include<iostream>
#include<stdlib.h>
using namespace std;

struct nodo{
    int dato;
    struct nodo *siguiente;
};

typedef nodo *Pila;

//Prototipos de Funciones:
void menu();
void push(Pila &, int);
void mostrarPila(Pila);
bool vacia(Pila);
void reemplazar(Pila &, int, int);

Pila p = NULL;
int main(){
    int dato, datoNew, datoOld;
    menu();

    cout<<"\n";
    system("pause");
    return 0;
}
```

```
//Función Menú:
void menu(){
    int dato, datoNew, datoOld, opc;

    do{
        cout<<"\t.:MENU:.\n";
        cout<<"1. Apilar Numero a la PILA\n";
        cout<<"2. Mostrar los Numeros de la PILA\n";
        cout<<"3. Reemplazar Numero en PILA\n";
        cout<<"4. Salir\n";
        cout<<"Opcion: "; cin>>opc;

        switch(opc){
            case 1: cout<<"Inserte un numero para Apilar: "; cin>>dato;
                    push(p, dato);
                    cout<<"\nNumero "<<dato<<" apilado...";
                    cout<<"\n";
                    system("pause");
                    break;
            case 2: cout<<"\nElementos en Pila: "<<endl;
                    if(p != NULL){
                        mostrarPila(p);
                    }
                    else{
                        cout<<"\nPila vacia..."<<endl;
                    }
                    cout<<"\n";
                    system("pause");
                    break;
            case 3: if(!vacia(p)){
                        cout<<"\nInserte el Numero que desea Reemplazar: "; cin>>datoOld;
                        cout<<"\nAhora, inserte el Nuevo Numero: "; cin>>datoNew;
                        reemplazar(p, datoNew, datoOld);
                    }
```

```
                    else{
                        cout<<"\nPila vacia..."<<endl;
                    }
                    cout<<"\nReemplazo logrado";
                    cout<<"\n";
                    system("pause");
                    break;
        }
        system("cls");
    }while(opc != 4);
}

//Función Push, para Apilar:
void push(Pila &p, int valor){
    Pila aux;
    aux = new(struct nodo);
    aux->dato = valor;
    aux->siguiente = p;
    p = aux;
}

//Función Mostrar Pila:
void mostrarPila(Pila p){
    Pila aux;
    aux = p;
    while(aux != NULL){
        cout<<"\n"<<aux->dato<<endl;
        aux = aux->siguiente;
   }
}

//Función para Comprobar el Estado de la Pila (Vacía o no):
bool vacia(Pila p){
    return (p == NULL ? true : false);
}
```

```
//Función Requerida para Reemplazar los Número de la Pila:
void reemplazar(Pila &p, int nuevo, int viejo){
    Pila aux = new nodo();
    aux = NULL;

    if(p == NULL){
        cout<<"La PILA esta vacia\n";
    }
    else{
        while(p != NULL){
            if(p->dato != viejo){
                push(aux, p->dato);
            }
            else{
                p->dato = nuevo;
                break;
            }
            p = p->siguiente;
        }
        while(aux != NULL){
            push(p, aux->dato);
            aux = aux->siguiente;
        }
    }
}
```

M: 2, U: 4, O: 4

4. Se dispone de un árbol binario de elementos de tipo entero. Escriba las funciones en C ++ que calculen:

a) La suma de sus elementos.

b) La suma de sus elementos múltiplos de 3.

Respuesta:

Código en C++:

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;

struct Nodo{
        int dato;
        Nodo *der;
        Nodo *izq;
};

Nodo *crearNodo(int);
void insertarNodo(Nodo *&, int);
int suma(Nodo *);
int sumaMultiplo(Nodo *);
void mostrarArbol(Nodo *, int);

Nodo *arbol = NULL;

int main(){
    int dato, contador = 0;
    char opc;

    do{
        cout<<"Inserte un numero: "; cin>>dato;
```

```
        insertarNodo(arbol, dato);

        cout<<"\nInsertar otro numero? S/N: "; cin>>opc;
        opc = toupper(opc);
        system("cls");
    }while(opc != 'N');

    cout<<"\nLa Sumatoria de los Numeros del Arbol es: "<<suma(arbol)<<endl;
    cout<<"\nLa Suma de los Numeros del Arbol, Multiplos de 3, es: "<<sumaMultiplo(arbol)<<endl;

    cout<<"\nMostrando el ARBOL completo:\n\n";
    mostrarArbol(arbol, contador);

    cout<<"\n";
    system("pause");
    return 0;
}

//Función Crear Nodo:
Nodo *crearNodo(int n){
    Nodo *nuevo_nodo = new Nodo();

    nuevo_nodo->dato = n;
    nuevo_nodo->der = NULL;
    nuevo_nodo->izq = NULL;

    return nuevo_nodo;
}

//Función Insertar:
void insertarNodo(Nodo *&arbol, int n){

    if(arbol == NULL){
        Nodo *nuevo_nodo = crearNodo(n);
```

```
        arbol = nuevo_nodo;
    }
    else{
        int valorRaiz = arbol->dato;
        if(n < valorRaiz){
            insertarNodo(arbol->izq, n);
        }
        else{
            insertarNodo(arbol->der, n);
        }
    }
}

//Función  Requerida de Sumatoria:
int suma(Nodo *arbol){
    int sumatoria = 0;

    if(arbol == NULL){
        sumatoria = 0;
    }
    else{
        sumatoria = arbol->dato + suma(arbol->der) + suma(arbol->izq);
    }

    return sumatoria;
}

//Función  Requerida de Sumatoria de Múltiplos:
int sumaMultiplo(Nodo *arbol){
    int sm = 0;

    if(arbol == NULL){
        sm = 0;
    }
    else{
```

```
        if((arbol->dato)%3 == 0){
            sm = arbol->dato + sumaMultiplo(arbol->der) + sumaMultiplo(arbol->izq);
        }
        else{
            sm = sumaMultiplo(arbol->der) + sumaMultiplo(arbol->izq);
        }
    }

    return sm;
}

//Función Mostrar de forma Completa:
void mostrarArbol(Nodo *arbol, int cont){
    if(arbol == NULL){
        return;
    }
    else{
        mostrarArbol(arbol->der, cont+1);
        for(int i=0; i<cont; i++){
            cout<<"   ";
        }
        cout<<arbol->dato<<endl;
        mostrarArbol(arbol->izq, cont+1);
    }
}
```

FIN DEL TRABAJO PRÁCTICO SUSTITUTIVO