CS1102 Programación Orientada a Objetos 1

Unidad 1: Estructuras de control

Asesoría - Semana 3

Asesores:

Plinio Avendaño	*plinio.avendano@utec.edu.pe*
Alonso Barrios	*alonso.barrios@utec.edu.pe*
Alex Loja	*alex.loja@utec.edu.pe*
Sebastian Peñaranda	*sebastian.penaranda@utec.edu.pe*

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Autores:

María Hilda Bermejo

1 Ejercicios

Ejercicio 1:

Se le pide crear una aplicación que dada una contraseña ingresada por el usuario, deberá de encriptar la contraseña de la siguiente manera:

- Cada letra deberá de ser pasada a su número ASCII correspondiente.

Restricciones:

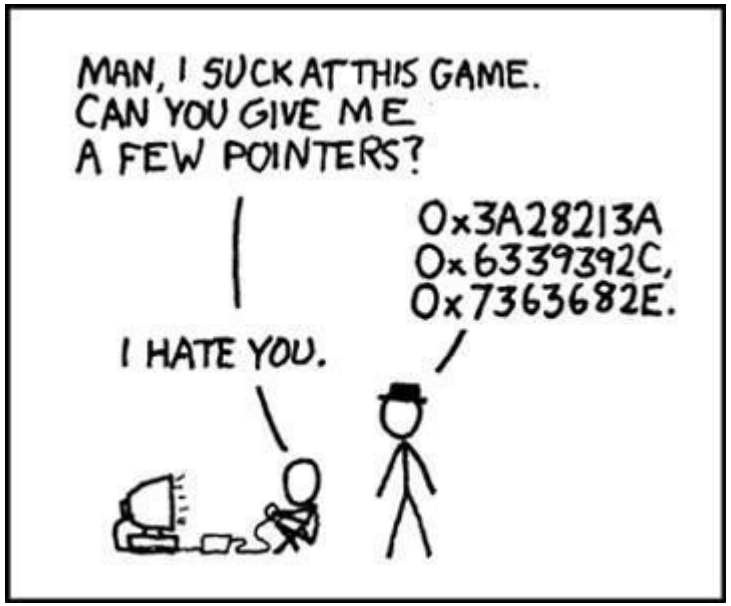
- No se permiten espacios en blanco.
- El usuario no podrá ingresar caracteres especiales.

Input:

hola31

Output:

1041111089731



Solución:

```cpp
#include <iostream>
#include <cctype>
#include <string>

using namespace std;

string encrypt(string*);

int main(int argc, char *argv[]) {
    string password = "";

    cout << "Password: ";
    cin >> password;

    cout << "Encrypted password: " << encrypt(&password) << endl;

    return 0;
}

string encrypt(string* password) {
    string encrypted_password = "";

    for(const auto& letter : *password) {
        if(!isdigit(letter)) {
            encrypted_password += to_string(int(letter));
        } else {
            encrypted_password += letter;
        }
    }

    return encrypted_password;
}
```

Ejercicio 2:

Se desea crear una aplicación que reciba un conjunto de notas (5) que se encargue de calcular lo siguiente:

- Promedio.
- Mayor nota (junto con su índice).
- Menor nota (junto con su índice).

Hint: Los punteros pueden funcionar como listas (python) o arrays (C++).

Input:

{18, 10.5, 14.6, 12, 19.4}

Output:

Promedio: 14.9

Mayor: 19.4

Menor: 10.5

Solución:

```cpp
#include <iostream>

using namespace std;

const int TOTAL_GRADES = 5;

void getData(double*&);

int main(int argc, char *argv[]) {
    double* grades = new double[100];

    getData(grades);

    double sum = 0;
    double greater = 0;
    double less = 10000;

    for(int i = 0; i < TOTAL_GRADES; i++) {
        sum += *(grades + i);

        if(greater < *(grades + i)) {
            greater = *(grades + i);
        }

        if(less > *(grades + i)) {
            less = *(grades + i);
        }
    }

    cout << "Average: " << sum / TOTAL_GRADES << endl;
    cout << "Greater: " << greater << endl;
    cout << "Less: " << less << endl;

    delete []grades;

    return 0;
}

void getData(double*& grades) {
    int i = 0;

    do {
        cin >> *(grades + i);
        i++;
    } while(i < TOTAL_GRADES);
}
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Ejercicio 3:

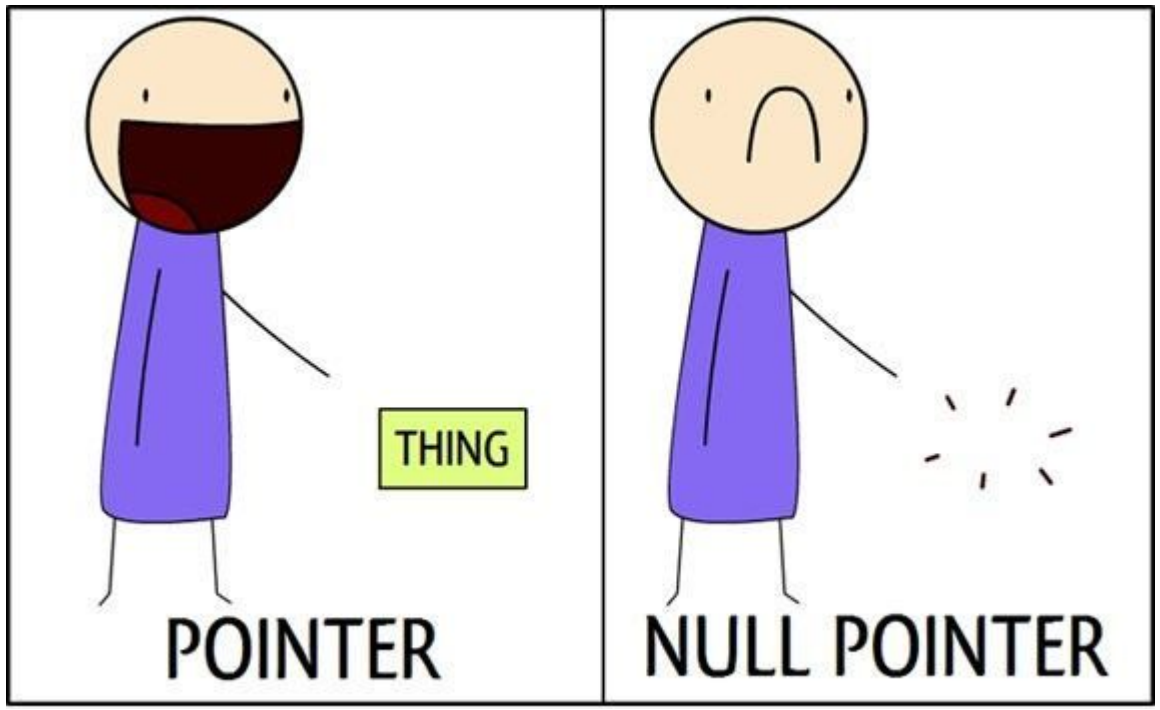
Implementar la función *insertionSort* que reciba como parámetro un puntero (que se usará como *array*). La función se encargará de ordenar el arreglo dado.
Referencia: Puedes ver el algoritmo en el siguiente video o link.

Input:

{5, 2, 1, 4, 3}

Output:

{1, 2, 3, 4, 5}



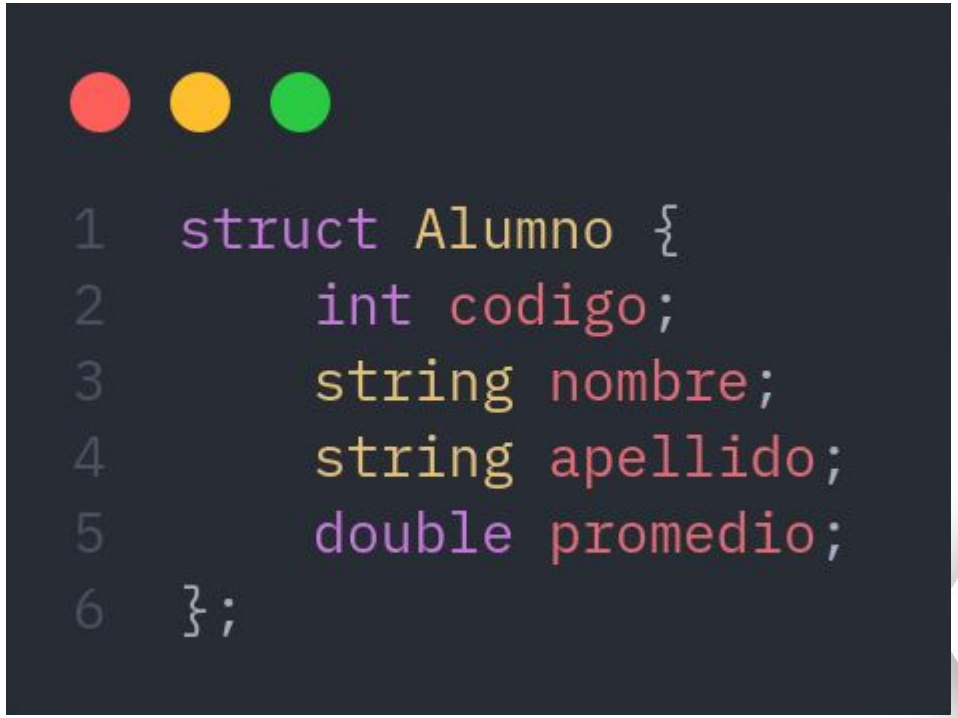
Solución:

```cpp
#include <iostream>

using namespace std;

void print(int*&, const int&);
void insertionSort(int*&, const int&);

int main(int argc, char *argv[]) {
    int size = 0;

    cout << "Insert number of elements: ";
    cin >> size;

    int* my_array = new int[size];

    for(int i = 0; i < size; i++) {
        cin >> *(my_array + i);
    }

    cout << "Original: ";
    print(my_array, size);

    insertionSort(my_array, size);
    cout << "Sorted: ";
    print(my_array, size);

    return 0;
}

void print(int*& my_array, const int& size) {
    cout << "{";

    for(int i = 0; i < size; i++) {
        cout << *(my_array + i) << ((i == size - 1) ? "" : ", ");
    }

    cout << "}" << endl;
}

void insertionSort(int*& my_array, const int& size) {
    int key = 0;
    int j = 0;

    for(int i = 1; i < size; i++) {
        key = *(my_array + i);
        j = i - 1;

        while(j >= 0 && *(my_array + j) > key) {
            *(my_array + j + 1) = *(my_array + j);
            j--;
        }

        *(my_array + j + 1) = key;
    }
}
```

2 Struct

¿Qué es un *Struct*?

- Es una colección de uno o más tipos de elementos denominados **atributos**.
- Cada atributo puede tener un diferente tipo de dato.
- A diferencia de las clases, los atributos son **públicos** por defecto.



UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

¿Cómo acceder a los atributos?

- Si el *struct* es **normal**, se usa el punto ‘ . ‘.

```
1  Alumno alumno1;
2  alumno1.codigo = 2;
```

- Si el *struct es* **puntero**, se usa una flecha ‘ -> ’.

```
1  Alumno* ptr_alumno = nullptr;
2  ptr_alumno→codigo = 2;
```

Grabaciones:

https://drive.google.com/drive/folders/1FpGdQuMi_rxOfIAwoS1M1B3MSNAKIQeK?usp=sharing

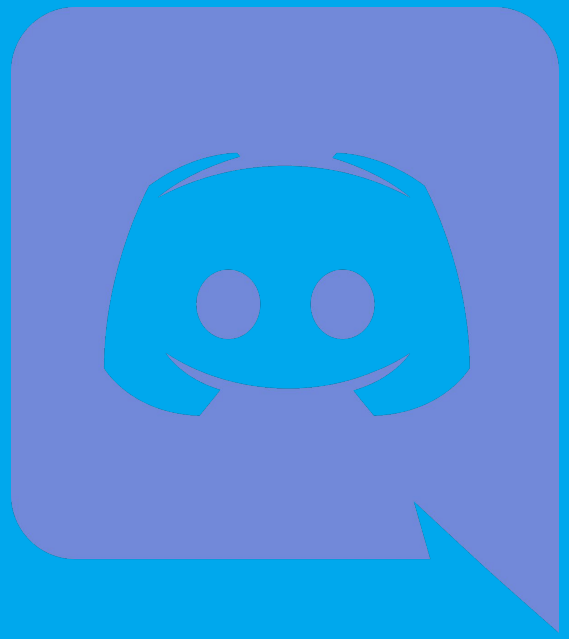
Repositorio:

https://github.com/alonso804/POO-I-Asesorias

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Discord server:

https://discord.gg/jADvs4GM4E



UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Gracias

UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA