



UNSA
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

“UNIVERSIDAD NACIONAL DE SAN AGUSTÍN”

**FACULTAD DE INGENIERÍA, PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA
COMPUTACIÓN**

CURSO:

Ciencias de la Computación - Grupo “B”

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Fabricio Huaquisto Quispe

REPOSITORIO:

<https://github.com/fhuaquisto21/EPCC-CCII>

Arequipa - Perú

2022

1. Implementar un programa que maneje un arreglo de estructuras para almacenar los nombres y las fechas de cumpleaños de sus n compañeros. Y debe mostrarse por pantalla quienes cumplen años en este mes.

```
#include <iostream>
struct FNacimiento {
    int day;
    int month;
    int year;
};
struct Persona {
    char* name;
    FNacimiento fechaNacimiento;
};

int main() {
    int monthToday = 7;
    int n;
    std::cout << "Cantidad de alumnos: ";
    std::cin >> n;
    Persona personas[n];

    for (int i = 0; i < n; ++i) {
        char* _name = new char[50];
        int _day, _month, _year;
        std::cout << "Nombre: ";
        std::cin >> _name;
        std::cout << "Día de nacimiento: ";
        std::cin >> _day;
        std::cout << "Mes de nacimiento: ";
        std::cin >> _month;
        std::cout << "Año de nacimiento: ";
        std::cin >> _year;

        Persona aux = {_name, _day, _month, _year};
        personas[i] = aux;
        std::cout << std::endl;
    }

    std::cout << std::endl;
    std::cout << "Alumnos que cumple años en Julio:" << std::endl;
    for (int i = 0; i < n; ++i) {
        if (personas[i].fechaNacimiento.month == monthToday) {
            std::cout << personas[i].name << std::endl;
        }
    }
    return 0;
}
```

fhuaquista: 13

→ ./a.out

Cantidad de alumnos: 3

Nombre: Fabricio

Día de nacimiento: 30

Mes de nacimiento: 3

Año de nacimiento: 2003

Nombre: Juan

Día de nacimiento: 12

Mes de nacimiento: 7

Año de nacimiento: 2002

Nombre: Mario

Día de nacimiento: 21

Mes de nacimiento: 7

Año de nacimiento: 2001

Alumnos que cumple años en Julio:

Juan

Mario

fhuaquista: 13

2. Implementar un programa que maneje un arreglo de estructuras que calcule la nota final del Ciencia de la Computación. El programa debe permitir el ingreso de cualquier cantidad de alumnos y para cada alumno, se podrá ingresar nombre, grupo, nota de la primera fase, segunda fase, tercera fase y proyecto final. El porcentaje de cada ítem es 15%, 20%, 25% y 40% respectivamente.

```
#include <iostream>
```

```
struct Nota {  
    int fase_1;  
    int fase_2;  
    int fase_3;  
    int proyecto;  
};
```

```
struct Alumno {  
    char* name;  
    char grupo;  
    Nota notas;  
};
```

```
int main() {  
    int n;  
    std::cout << "Cantidad de alumnos: ";  
    std::cin >> n;  
    Alumno alumnos[n];  
  
    for (int i = 0; i < n; ++i) {  
        char* _name = new char[50];  
        char _grupo;  
        int _fase_1, _fase_2, _fase_3, _proyecto;  
        std::cout << "Nombre: ";  
        std::cin >> _name;  
        std::cout << "Grupo: ";  
        std::cin >> _grupo;  
        std::cout << "Nota fase 1: ";  
        std::cin >> _fase_1;  
        std::cout << "Nota fase 2: ";  
        std::cin >> _fase_2;  
        std::cout << "Nota fase 3: ";  
        std::cin >> _fase_3;  
        std::cout << "Nota proyecto: ";  
        std::cin >> _proyecto;
```

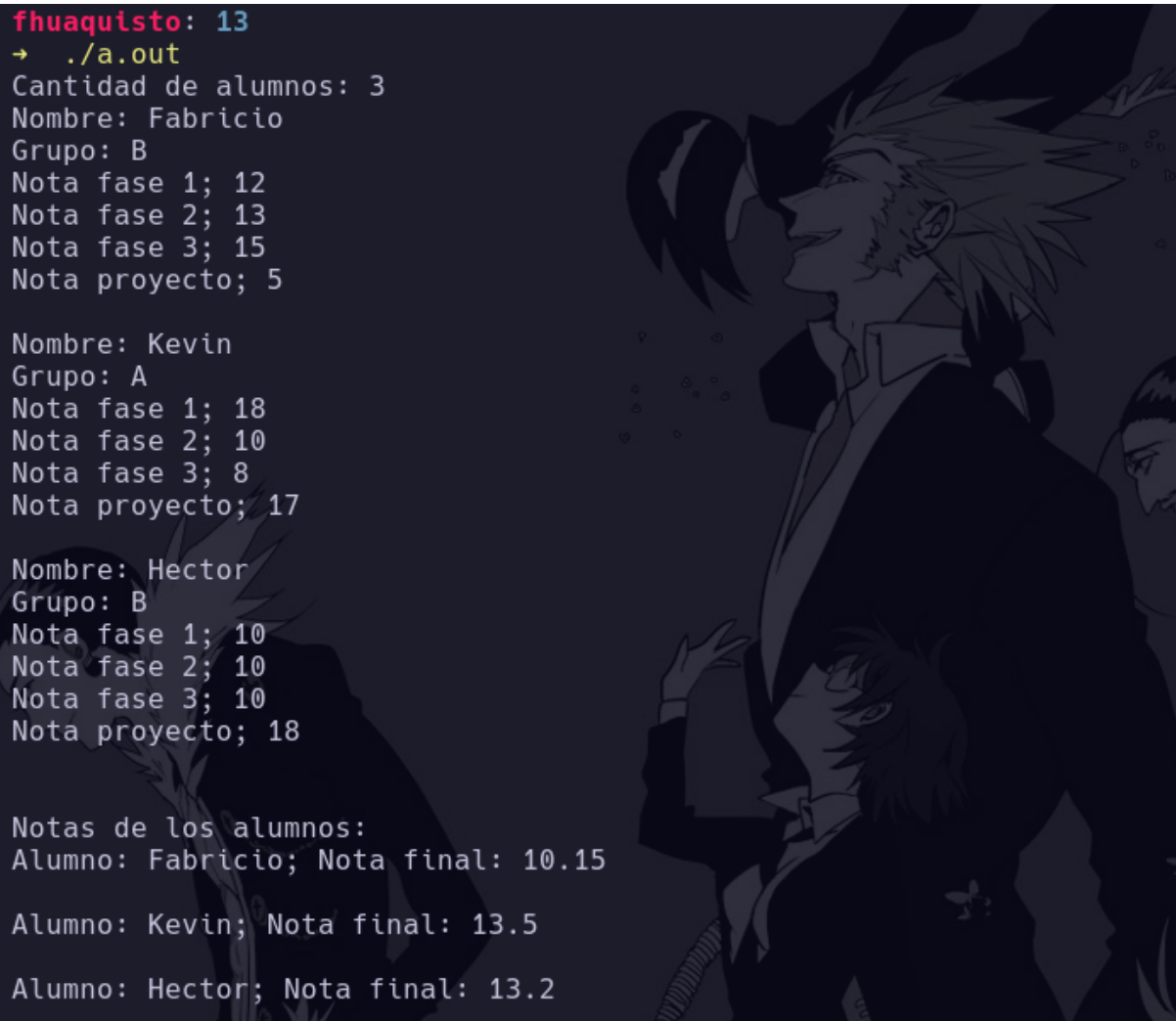
```
        Alumno aux = {_name, _grupo, _fase_1, _fase_2, _fase_3, _proyecto};  
        alumnos[i] = aux;
```

```

        std::cout << std::endl;
    }

    std::cout << std::endl;
    std::cout << "Notas de los alumnos:" << std::endl;
    for (int i = 0; i < n; ++i) {
        std::cout << "Alumno: " << alumnos[i].name;
        float notaFinal = (alumnos[i].notas.fase_1 * .15) +
(alumnos[i].notas.fase_2 * .2) + (alumnos[i].notas.fase_3 * .25) +
(alumnos[i].notas.proyecto * .4);
        std::cout << "; Nota final: " << notaFinal << std::endl;
        std::cout << std::endl;
    }
    return 0;
}

```



A terminal window showing the execution of a C++ program. The background features a dark anime-style illustration of several characters. The terminal output is as follows:

```

fhuaquisto: 13
→ ./a.out
Cantidad de alumnos: 3
Nombre: Fabricio
Grupo: B
Nota fase 1; 12
Nota fase 2; 13
Nota fase 3; 15
Nota proyecto; 5

Nombre: Kevin
Grupo: A
Nota fase 1; 18
Nota fase 2; 10
Nota fase 3; 8
Nota proyecto; 17

Nombre: Hector
Grupo: B
Nota fase 1; 10
Nota fase 2; 10
Nota fase 3; 10
Nota proyecto; 18

Notas de los alumnos:
Alumno: Fabricio; Nota final: 10.15

Alumno: Kevin; Nota final: 13.5

Alumno: Hector; Nota final: 13.2

```

3. Implemente un programa que maneje un arreglo de estructuras que solicite el nombre, edad y talla de 10 jugadores, debe mostrar por pantalla los que son menores de 20 años y tienen una talla mayor a 1,70 mts de altura.

```
#include <iostream>

struct Jugador{
    char* nombre;
    int edad;
    float talla;
};

int main() {
    Jugador jugadores[10];

    for (int i = 0; i < 10; ++i) {
        char* _nombre = new char[50];
        int _edad;
        float _talla;
        std::cout << "Nombre: ";
        std::cin >> _nombre;
        std::cout << "Edad: ";
        std::cin >> _edad;
        std::cout << "Talla: ";
        std::cin >> _talla;

        Jugador aux = {_nombre, _edad, _talla };
        jugadores[i] = aux;
        std::cout << std::endl;
    }

    std::cout << std::endl;
    std::cout << "Jugadores menores de 20 y de talla +1.70:" << std::endl;
    for (int i = 0; i < 10; ++i) {
        if (jugadores[i].edad < 20) {
            if (jugadores[i].talla > 1.7) {
                std::cout << "Jugador: " << jugadores[i].nombre << std::endl;
            }
        }
    }
    return 0;
}
```

Nombre: Fabricio
Edad: 19
Talla; 1.71

Nombre: Mario
Edad: 21
Talla; 1.7

Nombre: Juan
Edad: 20
Talla; 1.7

Nombre: Kevin
Edad: 19
Talla; 1.69

Nombre: Edgar
Edad: 15
Talla; 1.72

Nombre: Pequeño
Edad: 12
Talla; 1.85

Nombre: Pedro
Edad: 20
Talla; 1.75

Nombre: Victor
Edad: 19
Talla; 1.72

Nombre: Salazar
Edad: 18
Talla; 1.6

Nombre: Bryan
Edad: 17
Talla; 1.73

Jugadores menores de 20 y de talla +1.70:

Jugador: Fabricio
Jugador: Edgar
Jugador: Pequeño
Jugador: Victor
Jugador: Bryan

fhuaquisto: 13



4. Implemente un programa que maneje un arreglo de estructuras que solicite nombre, sexo y sueldo de los empleados de una empresa y debe mostrar por pantalla el menor y mayor sueldo.

```
#include <iostream>

struct Empleado{
    char* nombre;
    char sexo;
    float sueldo;
};

int main() {
    int n;
    std::cout << "Cantidad de empleados: ";
    std::cin >> n;
    Empleado empleados[n];

    for (int i = 0; i < n; ++i) {
        char* _nombre = new char[50];
        char _sexo;
        float _sueldo;
        std::cout << "Nombre: ";
        std::cin >> _nombre;
        std::cout << "Sexo [M, H]: ";
        std::cin >> _sexo;
        std::cout << "Sueldo: ";
        std::cin >> _sueldo;

        Empleado aux = {_nombre, _sexo, _sueldo };
        empleados[i] = aux;
        std::cout << std::endl;
    }

    std::cout << std::endl;
    float minSueldo = empleados[0].sueldo, maxSueldo = empleados[0].sueldo;
    for (int i = 0; i < n; ++i) {
        if (empleados[i].sueldo > maxSueldo) {
            maxSueldo = empleados[i].sueldo;
        }
        if (empleados[i].sueldo < minSueldo) {
            minSueldo = empleados[i].sueldo;
        }
    }

    std::cout << "El peor sueldo: " << minSueldo << std::endl;
    std::cout << "El mejor sueldo: " << maxSueldo << std::endl;
```



```
    return 0;  
}
```

```
fhuaquistto: 13  
→ ./a.out  
Cantidad de empleados: 4  
Nombre: Fabricio  
Sexo [M, H]: H  
Sueldo; 1625.5  
  
Nombre: Juan  
Sexo [M, H]: H  
Sueldo; 1450  
  
Nombre: María  
Sexo [M, H]: M  
Sueldo; 1567.89  
  
Nombre: Juana  
Sexo [M, H]: M  
Sueldo; 1449.9  
  
El peor sueldo: 1449.9  
El mejor sueldo: 1625.5  
fhuaquistto: 13
```