

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**  
**PROGRAMACIÓN**

**PROF. ING. GUIDO ACOSTA**

**Laboratorio – Registros**

**Ejercicios – Nivel Inicial**

1. Escribir un programa que registre los datos de 5 personas en un vector. Por cada persona se deberá registrar: nombre, apellido, edad, sexo, teléfono. Imprimir todos los datos.
2. Escribir un programa que registre los datos de 5 libros en un vector. Por cada libro se deberá registrar: título, autor, año. Imprimir todos los datos.

**Ejercicios – Nivel Intermedio**

3. Escribir un programa que registre 10 fechas en un vector. La fecha deberá ser de tipo estructura donde los campos serán día, mes y año. Luego ingresar una fecha y verificar si la misma se encuentra en el vector.
4. Escribir un programa que lea los datos de N alumnos (nombre, apellido, edad, nombre carrera) y las nota de cada alumno en dos materias (nombre, nota). Imprimir el nombre del alumno con el mejor promedio entre las dos materias.
5. Escribir un programa que pida los datos de N alumnos (nombre, edad, coeficiente intelectual). Imprimir el nombre del alumno con el mayor coeficiente intelectual y el promedio del grupo.
6. Crear un programa que declara la siguiente estructura

```
struct alumno
{
    char nombre[25];
    char apellido[25];
    int nro_documento;
    int notaPrimerParcial;
    int notaSegundoParcial;
};
```

El programa deberá permitir la carga de datos para 10 alumnos.

7. Modificar la estructura del ejercicio anterior para agregar los siguientes datos del alumno
  - a. Promedio ponderado
  - b. Nota examen final
  - c. Calificación

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**  
**PROGRAMACIÓN**

**PROF. ING. GUIDO ACOSTA**

El programa además de leer los datos del ejercicio anterior, deberá permitir la lectura de la nota examen final. El programa deberá calcular el promedio ponderado como el promedio de las notas parciales. Luego deberá calcular la nota final utilizando la siguiente fórmula:  $\text{promedio ponderado} * 0.4 + \text{nota final} * 0.6$

8. Modificar el programa anterior para imprimir el nombre y número de documento del alumno con la mejor calificación. Si existe más de un alumno con la mejor calificación, se deberá imprimir el nombre de todos los alumnos con la mejor calificación.
9. Modificar el programa anterior para agregar un menú con las siguientes opciones:
  - a. Cargar alumnos (deberá cargar los datos para 10 alumnos)
  - b. Imprimir el listado de todos los alumnos (Se deberá imprimir el nombre y el número de documento)
  - c. Imprimir alumnos con la mejor calificación
  - d. Imprimir la nota más alta y la más baja
  - e. Imprimir el promedio de la notas
  - f. Imprimir la lista de alumnos que aprobaron
  - g. Imprimir la lista de alumnos que reprobaron

10. Crear un programa que declare la siguiente estructura

```
struct persona
{
    char nombre[25];
    char apellido[25];
    int nro_documento;
};
```

Modificar la estructura alumno para que contenga la estructura persona. Modificar las funcionalidades del ejercicio 4 para tener en cuenta la nueva estructura.

### **Ejercicios – Nivel Avanzado**

11. Escribir un programa que lea los datos de 10 estudiantes (nombre, edad, sexo). Cada estudiante está registrado en las 3 materias (nombre, nota, unidades) del

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**  
**FACULTAD POLITÉCNICA**  
**PROGRAMACIÓN**

**PROF. ING. GUIDO ACOSTA**

semestre. Por cada unidad (nombre, puntaje) se obtienen puntajes y existen 3 unidades por materia. La nota de cada materia se obtiene a partir del promedio de los puntajes de las unidades. Imprimir los nombres de los alumnos con sus respectivas notas en las diferentes materias.

12. Pac es un muñeco experto en aritmética que le gusta buscar los caminos con mayores resultados. Para demostrar su habilidad, Pac recorre las matrices. Puede realizar recorridos verticales y horizontales. Realizar las siguientes tareas para que Pac haga su trabajo:

I. Declarar una matriz 6x6 y cargar los campos de los registros en cada celda. La matriz de abajo muestra un ejemplo.

II. Solicitar la posición donde será colocado Pac; fila y columna.

III. A partir de la posición inicial, Pac deberá realizar el recorrido a los cuatros lados buscando el resultado mayor; arriba, abajo, izquierda y derecha.

IV. Al finalizar el recorrido, el programa deberá indicar el camino que obtuvo el mayor resultado.

V. El programa deberá brindar la posibilidad de realizar varias veces el proceso. El programa termina cuando el usuario ingresa la posición -1, -1.

**Ejemplo**

	0	1	2	3	4	5
0	num1: 5 num2: 8 operación: *	num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: *	num1: 10 num2: 2 operación: /	num1: 5 num2: 8 operación: -	num1: 5 num2: 8 operación: +
1	num1: 2 num2: 18 operación: -	num1: 35 num2: 5 operación: /	num1: 5 num2: 8 operación: +	num1: 1 num2: 8 operación: +	num1: 5 num2: 8 operación: -	num1: 64 num2: 8 operación: /
2	num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: *	num1: 5 num2: 8 operación: *	num1: 45 num2: 5 operación: /	num1: 5 num2: 8 operación: -	num1: 5 num2: 8 operación: -
3	num1: 5 num2: 8 operación: *	num1: 25 num2: 5 operación: /	num1: 12 num2: 3 operación: /	num1: 5 num2: 8 operación: *	num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: +
4	num1: 49 num2: 7 operación: /	num1: 5 num2: 8 operación: -		num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: -	num1: 10 num2: 2 operación: /
5	num1: 5 num2: 8 operación: *	num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: +	num1: 5 num2: 8 operación: -	num1: 15 num2: 3 operación: /	num1: 5 num2: 8 operación: *