



Algoritmos y Estructuras de Datos II

Serie de Repaso

OBJETIVOS:

Realizar una práctica de repaso sobre los conceptos básicos de programación como ser: programa, tipos de datos, estructuras de control, procedimientos y funciones, arreglos y archivos.

METODOLOGÍA

- El alumno deberá resolver individualmente los ejercicios propuestos.
- El alumno deberá codificar las soluciones en el lenguaje de programación C.
- Realizar consultas a través del canal de **slack** correspondiente a su comisión ó del aula virtual de la asignatura.

COMPETENCIAS

- Identificar, formular y resolver problemas mediante programación.
- Utilizar de manera efectiva técnicas y herramientas de aplicación para desarrollar software.
- Desempeñarse de manera efectiva en equipos de trabajo.
- Aprender en forma continua, autónoma y de manera colaborativa.

DURACIÓN

Según planificación de la asignatura se deberá utilizar para la resolución de los ejercicios de esta serie de repaso 2 clases prácticas.

CONSIGNA:

Resolver la siguiente ejercitación, teniendo en cuenta que los ejercicios propuestos que siguen a continuación, se deberán resolver utilizando procedimientos o funciones según corresponda.

EJERCICIOS PROPUESTOS

1. Realizar un programa para calcular la remuneración semanal de los empleados de una fábrica. El valor de la hora laboral es de \$350 hasta 35 hs trabajadas, por encima de este valor se considera hora extra y el valor es de \$525. Los datos que se ingresan son: *apellido y nombre*, y *cantidad de horas trabajadas en la semana*. Se debe incluir una función **salarioSemanal** que reciba la cantidad de horas trabajadas y retorne el salario del trabajador. El programa debe permitir realizar el cálculo tantas veces como el usuario lo desee.

2. Escribir una función **generarVectorPares** que genere un vector con los números pares comprendidos entre 1 y 100. Luego, escribir una función **maximo** que reciba un vector como parámetro y retorne el máximo elemento del arreglo. Probar la función **maximo** en la función principal, utilizando el vector de números pares generado anteriormente.
3. Escribir una función **cargarVector** que permita almacenar valores que el usuario introduce por teclado. Luego escribir la función **sumaDeVectores** que reciba dos vectores A y B como parámetros de entrada, y permita visualizar la suma de cada elemento de A y B. Hacer uso de estas funciones invocándolas desde la función **main**.
4. Realizar un programa que permita visualizar una pirámide de asteriscos. Para ello se debe solicitar el ingreso de un número entero que representa la cantidad de filas que tendrá la pirámide. La primera fila de la pirámide debe tener un solo asterisco centrado, en la 2da fila, el doble de la primera más uno, y así sucesivamente. Por ejemplo, si se introduce el 5, el resultado sería:

```
Ingresar cantidad de filas de la piramide: 5
      *
     ***
    *****
   *********
  ***********
```

Se debe incluir una función **dibujarLinea** que reciba como parámetro la cantidad de asteriscos de la línea por dibujar, y una función **dibujarPiramide** que reciba como parámetro la cantidad de filas de la pirámide.

5. Realizar un programa en C que permita grabar un archivo de datos (binario) con los siguientes campos: *Número de Cuenta, Apellido y nombre, Código de movimiento (1-Depósito 2-Extracción), Importe*.
6. Realizar un programa en C que permita leer el archivo generado en el punto anterior y emita un listado de todas las cuentas. Obtener e informar al final la cantidad de depósitos y extracciones y el importe total de cada tipo de movimiento. Detalle del informe:

NRO. DE CUENTA	APELLIDO Y NOMBRE	MOVIMIENTO	IMPORTE
11111	María López	DEPÓSITO	\$5.500
11111	María López	EXTRACCIÓN	\$1.800
2222	Juan Acosta	EXTRACCIÓN	\$4.000
3333	Mariana Pérez	DEPÓSITO	\$10.000
4444	Agustina Acosta	EXTRACCIÓN	\$7.000

CANTIDAD DE DEPÓSITOS: 2

IMPORTE TOTAL: \$15.500

CANTIDAD DE EXTRACCIONES: 3

IMPORTE TOTAL: \$12.800