

Ejercicios de práctica Estructuras Secuenciales y Condicionales

1. Analice el siguiente código fuente, dado en lenguaje C/C++, y determine su propósito. Reemplace los símbolos 😊😊😊😊 por los mensajes adecuados en cada caso. Escriba el programa, guárdelo como ejercicio1.cpp y luego compile y ejecute.

```
// librerias que permiten utilizar las funciones del lenguaje
#include <iostream>
#include <stdlib.h>
using namespace std;
// programa principal
main ()
{ int num1,num2;
  cout << "Ingrese primer valor:";
  cin >> num1;
  cout << "Ingrese segundo valor:";
  cin >> num2;
  if (num2 != 0)
  {
    if (num1>num2 && num2>0)
    { cout << "😊😊😊😊 " << num1/num2 << endl;
      cout << "😊😊😊😊 " << num1%num2 << endl;
    }
    else
      cout << "😊😊😊😊" << endl;
  }
  else
    cout << "😊😊😊😊" << endl;
  system("pause");
}
```

2. Analice el siguiente código fuente, dado en lenguaje C/C++, y determine su propósito. Escriba el programa, guárdelo como ejercicio2.cpp y luego compile y ejecute. ¿Para qué sirven las funciones *toupper* y *tolower*?

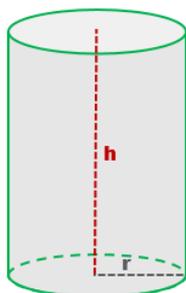
```
#include <iostream>
#include <stdlib.h>

using namespace std;
main ()
{ char letra;
  cout << "Ingrese caracter:";
  cin >> letra;
  if (letra>='a' && letra <='z')
  { letra=toupper(letra);
    cout << "Salida: " << letra << endl;
  }
  else
    if (letra>='A' && letra<='Z')
    { letra=tolower(letra);
      cout << "Salida: " << letra << endl;
    }
    else
      cout << "Salida: " << letra << endl;
    system("pause");
}
```

3. Analice el siguiente código fuente, dado en lenguaje C/C++, y determine su propósito. Escriba el programa, guárdelo como ejercicio3.cpp y luego compile y ejecute.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
main ()
{ char letra;
  cout << "Ingrese letra:";
  cin >> letra;
  letra=tolower(letra);
  switch (letra)
  {
  case 'a': cout << "VOCAL a/A" << endl;
            break;
  case 'e': cout << "VOCAL e/E" << endl;
            break;
  case 'i': cout << "VOCAL i/I" << endl;
            break;
  case 'o': cout << "VOCAL o/O" << endl;
            break;
  case 'u': cout << "VOCAL u/U" << endl;
            break;
  default: if (letra>'a' && letra <='z')
            cout << "ES UNA CONSONANTE" << endl;
            else
            cout << "ES UN SIMBOLO" << endl;
  }
  system("pause");
}
```

4. Un cilindro circular recto es un cuerpo geométrico que tiene dos bases circulares y un eje perpendicular a ambas bases. El eje es también la altura del cilindro. Sabiendo esto, diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la superficie y el volumen de un cilindro circular recto. Considere que los datos de altura (h) y radio (r) son ingresados por el usuario.



h: altura

r: radio

$$\text{Sup. base} = \pi \times \text{radio}^2$$

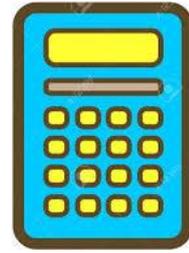
$$\text{Sup. lateral} = 2 \times \pi \times \text{radio} \times \text{altura}$$

$$\text{Superficie cilindro} = 2 \times \text{sup. base} + \text{sup. lateral}$$

$$\text{Volúmen cilindro} = \text{sup. base} \times \text{altura}$$

5. El jefe de programadores solicita agregar controles al algoritmo del ítem 2. Estos controles deben verificar que los datos ingresados sean válidos para el problema, indicando mediante mensajes las situaciones de error. ¿Podrías agregarlos?

6. Diseñe un algoritmo que simule el funcionamiento de una calculadora sencilla. Esta calculadora debería ser capaz de resolver la suma, la resta, el producto, la división real, la división entera, el resto y potencia de 2 valores. Para ello, el usuario proporcionará los valores a operar y un símbolo (+, -, *, /, \, %, ^) que indique la operación a realizar. ¿Puedes proponer 2 versiones (algoritmos) para resolver este problema? La solución debe incluir los controles necesarios para realizar correctamente las operaciones.



7. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que presente el siguiente menú de un cajero automático:

```
**** MENU ATM ****
E) Extracción
D) Depósito
T) Transferencia
C) Consulta
S) Salir
*****
Seleccione una opción:
```

Considere que el algoritmo, antes de mostrar el menú, debe solicitar al usuario los siguientes datos: nombre, apellido, tipo de cuenta (caja de ahorro, cuenta corriente) y saldo actual (un valor positivo). Al ejecutar las operaciones debe verificarse que la operación sea posible (saldo suficiente) y que se verifiquen las siguientes restricciones:

- las extracciones tienen un límite diario: \$50.000 pesos (caja de ahorro) y \$80.000 (cuenta corriente)
- los depósitos no pueden exceder los \$70.000 (caja de ahorro y cuenta corriente) debiendo notificarse al cliente que debe dirigirse a las cajas del banco para realizar la operación
- las transferencias no pueden superar los \$150.000 para cajas de ahorro y los \$250.000 para cuentas corrientes
- la operación de consulta debe mostrar el titular de la cuenta (apellido y nombre), tipo de cuenta, su saldo actual y el monto de extracción anticipada (20% del saldo actual para cajas de ahorro y 30% para cuentas corrientes).

Además, ante opciones incorrectas debe presentarse un mensaje de error.

8. Escriba un enunciado factible de ser solucionado mediante programación y en especial con estructuras secuenciales y condicionales. Codifique su solución.