

INSO - Introducción a la programación I – Curso 2024 - 2025

Ejercicio Evaluable 2

La práctica 1 estará formada por los siguientes 4 ejercicios:

- A. Práctica las tablas de multiplicar
- B. Incrementa hasta 100
- C. Calculadora de Potencia con Validación de Exponente
- D. Suma recursiva de los dígitos impares

A continuación, encontrarás los detalles de cada enunciado para escribir el código correspondiente. En todos ellos debes realizar el análisis del problema que se plantea, estudiando los datos de entrada que necesitas, plantea a continuación las sentencias de ejecución en base a las operaciones que debes realizar y finalmente muestra el resultado.

Se valorará si se ejecuta correctamente, así como la estructura general, la claridad y la legibilidad. Recuerda siempre utilizar identificadores que sean adecuados y el uso de comentarios.

Se valorará la simplicidad de las estructuras de control utilizada para implementar los algoritmos que resuelvan las necesidades de los programas. También debáis utilizar las funciones que se os sugiere en los enunciados

Ejercicio A. Practica las tablas de multiplicar

El siguiente ejercicio pretende crear un programa para practicar las tablas de multiplicar. Las instrucciones del juego son las siguientes:

- El programa solicita al usuario la tabla mayor incluida (mínimo 2)
- Habrá 10 rondas de preguntas con operandos elegidos al azar entre el 2 y el máximo solicitado anteriormente.
- El usuario tiene 3 vidas
- Si no introduce el resultado correcto resta una vida
- Si introduce el resultado correcto gana un punto
- El juego termina cuando se acaban las 10 rondas o cuando pierde todas las vidas.
- Al final de la partida deberá indicar cuantos puntos ha conseguido.

Se deberán implementar las siguientes funciones:

```
//Pide el maximo numero a multiplicar al usuario,  
//repite la operacion hasta que el usuario indique un  
//numero valido. El numero debe ser mayor que 1.  
int obtenerMaximo();  
  
// obtiene dos numeros aleatorios entre 2 y el maximo incluido  
// y los devuelve como parametros por referencia  
void obtenNumeros(int maximo, int *ope1, int *ope2);  
  
// genera una ronda, devuelve 0 si el usuario no ha acertado,  
// uno si todo va bien  
int rondaCorrecta(int maximo);
```

Ejemplo de ejecución:

```
Introduce el maximo:12  
4 * 2 :8  
4 * 7 :34  
....Resultado incorrecto, vidas: 2  
12 * 8 :96  
2 * 5 :10  
9 * 11 :99  
9 * 8 :76  
....Resultado incorrecto, vidas: 1  
5 * 6 :98  
....Resultado incorrecto, vidas: 0
```

```
Game Over  
4 puntos
```

Ejercicio B- Incremento hasta 100

Se pretende realizar un juego en el que el usuario, empezando desde el 1, debe llegar exactamente al número 100 siguiendo ciertas normas. Para ello el programa debe guiar al usuario y mostrar el progreso en cada paso.

Normas:

- En cada turno el usuario puede incrementar su número en 5 ó 3 unidades o puede multiplicarlo por 2.
- Existe un límite de 8 operaciones para alcanzar el número 100.

Condiciones de finalización del juego:

- Si el usuario alcanza exactamente el 100 el programa debe mostrar un mensaje de éxito y acabar.
- Si el usuario supera el 100 o utiliza más de las operaciones permitidas el programa debe mostrar un mensaje de derrota y terminar.
- Debe mostrarse al usuario una sugerencia cuando su número sea superior a 50 para que use la opción sumar y evitar superar el 100.

Se tienen que implementar las siguientes funciones:

```
// Debe mostrar las instrucciones del juego al principio de la partida
void mostrarInstruccionesIniciales();
// Debe mostrar el menú de elección de operación
void mostrarMenu(int numero, int operaciones);
// Debe realizar la operación elegida y devolver su resultado
int realizarOperacion(int numero, int opcion);
```

Ejemplo de ejecución:

```
Operacion 7:
1. Sumar 5
2. Sumar 3
3. Multiplicar por 2
Elige tu movimiento (1, 2, o 3): 1
Sugerencia: Estas cerca de 100, intenta usar sumas en lugar de multiplicar.
Numero actual: 99
Operacion 8:
1. Sumar 5
2. Sumar 3
3. Multiplicar por 2
Elige tu movimiento (1, 2, o 3): 2
Te has pasado de 100. Has perdido el juego.
```

```
Debes alcanzar exactamente el valor 100 desde el valor inicial de 1.
Puedes sumar 5, sumar 3 o multiplicar por 2 en cada movimiento.
Tienes un maximo de 20 movimientos. Buena suerte!
```

```
Numero actual: 1
Operacion 1:
1. Sumar 5
2. Sumar 3
3. Multiplicar por 2
Elige tu movimiento (1, 2, o 3): 1
Numero actual: 6
Operacion 2:
1. Sumar 5
2. Sumar 3
3. Multiplicar por 2
Elige tu movimiento (1, 2, o 3): 1
Numero actual: 11
```

Ejercicio C. Calculadora de Potencia con Validación de Exponente

El objetivo de este ejercicio es crear un programa que calcule la potencia de un número entero elevado a un exponente entero positivo. El programa debe solicitar al usuario un número base y un exponente, y mostrar el resultado solo si el exponente es positivo.

Normas:

1. La función `calcular_potencia` debe realizar la validación del exponente para asegurarse de que sea positivo.
2. Si el exponente es positivo, `calcular_potencia` realiza el cálculo de la potencia y retornar 1 para indicar que la operación fue exitosa. El resultado debe almacenarse en el puntero `*resultado`.
3. Si el exponente no es positivo, `calcular_potencia` debe retornar 0.
4. La función principal deberá mostrar el resultado o un mensaje de error en función del valor retornado por `calcular_potencia`.

Condiciones:

1. Si el cálculo se realiza correctamente, el programa debe mostrar el resultado de la potencia.
2. Si el exponente es menor o igual a 0, el programa debe mostrar un mensaje de error: "Error: el exponente debe ser positivo".

Se tiene que implementar la siguiente función:

```
// Esta función:  
// 1. Calcula la potencia de la base elevada al exponente usando un  
// bucle y asigna el resultado a la variable resultado.  
// 2. Valida si el exponente es positivo. Devuelve un 1 si el cálculo  
// se realiza correctamente o un 0 si el exponente no es positivo  
int calcular_potencia(int base, int exponente, int *resultado)
```

Ejemplo de ejecución:

Caso correcto:

```
Ingrese la base: 3  
Ingrese el exponente: 4  
Resultado: 81
```

Caso incorrecto:

```
Ingrese la base: 3  
Ingrese el exponente: -2  
Error: el exponente debe ser positivo.
```

Ejercicio D Suma recursiva de los dígitos impares

Realiza un programa que utilice una función recursiva para sumar los dígitos de posición impar (contando como posición 1 el dígito menos significativo) de un número entero.

El programa debe funcionar tanto para números positivos como negativos.

Se tiene que implementar la siguiente función:

```
// Función que de manera recursiva calcula la suma
// Como condición base si el número es igual a 0 la suma es 0
// La llamada recursiva se debe hacer descartando los dígitos pares
int sumaDigitosImpares(int num);
```

Ejemplo de ejecución:

Numero positivo

```
Introduce un numero: 12345_
La suma es 9
```

Número negativo

```
Introduce un numero: -12345
La suma es 9
```