

1. Diseñe e implemente un programa que solicite al usuario un número (en base 10) y una nueva base (en el intervalo [2,16]), e imprima la representación de ese número usando la base ingresada.

Para coeficientes entre 10 y 15 utilice los caracteres entre 'A' y 'F'.

Ejemplo:

284 en base 16:	11C
284 en base 4:	10130
284 en base 8:	434
284 en base 13:	18B

2. Modifique el programa anterior y encapsule el algoritmo en una función que reciba el valor y la nueva base, siendo la salida todavía realizada en el dispositivo estándar de salida.
3. Genere un archivo header con el prototipo de la función realizada en el ejercicio anterior y separe el main en otro archivo. Compile y corra nuevamente.
4. Realizar un programa que funcione como filtro de comentarios, definiendo como comentario a todo aquello colocado entre paréntesis (con posibilidad de comentarios anidados). Por ejemplo si se ingresa: "Juan (mi mejor amigo) llega hoy de viaje" la salida sería "Juan llega hoy de viaje".
5. Implemente las siguientes funciones de manejo de bits:

```
// retorna la cantidad de bits en 1 presentes
// en la representación de 'value'
int numOnes(int value);

// devuelve el valor 'value', con el bit 'bit' en 1
int setBit(int value, int bit);

// devuelve el valor 'value', con el bit 'bit' en 0
int resetBit(int value, int bit);

// chequea si el bit 'bit' está en 1
bool testBit(int value, int bit);
```

6. Implementar la función StrStr que reciba dos arreglos de caracteres y busque sobre el primero de ellos por la ocurrencia del segundo.

Prototipo: `int StrStr(char s1[], char s2[]);`

Como retorno, StrStr deberá retornar el índice del elemento sobre `s1` donde comienza la ocurrencia de `s2`. Si `s2` no aparece en `s1`, la función deberá retornar -1.

Ejemplos:

`StrStr("JUAN ESTA CASADO CON MARIA", "ASADO")` retornará 11.

`StrStr("ABCDE", "BCE")` retornará -1.

7. Escribir una función que reciba como parámetro una cadena de caracteres que puede comenzar con espacios en blanco, y los elimine desplazando los caracteres útiles hacia la izquierda. (operación "left-trim").

Prototipo: `void left_trim(char phrase[]);`

8. El algoritmo de Euclides permite calcular el máximo común divisor entre 2 números en forma muy eficiente. Se basa en que si tenemos 2 números enteros A y B, con $A \geq B$, el MCD entre A y B es equivalente al MCD entre B y el resto de la división entera entre A y B ($\text{MCD}(A, B) == \text{MCD}(B, A \% B)$). (Hint: el MCD entre cualquier número X y cero es X).
9. Se desea averiguar de cuántas formas posibles se puede llenar un rectángulo de $2 \times N$ casilleros utilizando fichas de dominó (de 2×1). En la figura se ve que para $N = 1$ hay una sola forma, para $N = 2$ hay 2 formas, para $N = 3$ hay 3 formas y para $N = 4$ hay 5 formas.

Escribir un programa que calcule la cantidad de formas de llenar con dominós un rectángulo de $2 \times N$ casilleros.

Hint: Descomponga el problema en combinaciones que empiecen con una ficha vertical y combinaciones que empiecen con fichas horizontales.

10. Definir una estructura `Complejo` que represente a un número complejo.

Implementar funciones para:

- Ingresar números complejos
- Imprimir
- Sumar
- Restar
- Multiplicar y
- Dividir números complejos

