Link to the project- <https://github.com/Tugamer89/Basic-Calculator/tree/main/Calculator>

This program implements a simple command-line calculator that can perform a variety of mathematical operations. It supports basic arithmetic, trigonometric functions, logarithms, and constants like PI and Euler's number.

**Class: calcs**

**Purpose**: Encapsulates a set of methods to perform various mathematical operations on a value.

**Members**:

* float value: Stores the current value on which operations are performed.

**Constructors**:

* calcs(): Default constructor that initializes value to 0.
* calcs(float n): Parameterized constructor that initializes value to n.

**Methods**:

* void plus(float n): Adds n to value.
* void minus(float n): Subtracts n from value.
* void mult(float n): Multiplies value by n.
* void div(float n): Divides value by n.
* void pow\_(float n): Raises value to the power of n.
* void sqrt\_(): Calculates the square root of value.
* void cos\_(): Calculates the cosine of value.
* void sin\_(): Calculates the sine of value.
* void tan\_(): Calculates the tangent of value.
* void acos\_(): Calculates the arccosine of value.
* void asin\_(): Calculates the arcsine of value.
* void atan\_(): Calculates the arctangent of value.
* void cbrt\_(): Calculates the cube root of value.
* void log\_(): Calculates the natural logarithm (base e) of value.
* void log2\_(): Calculates the logarithm base 2 of value.
* void log10\_(): Calculates the common logarithm (base 10) of value.

**Function: help**

**Purpose**: Displays a help sheet listing the supported operators and their corresponding mathematical functions.

**Steps**:

1. Clears the screen.
2. Prints the help sheet to the console.
3. Pauses the system to allow the user to read the help sheet.

**Function: isOperator**

**Purpose**: Checks if a character is a valid operator.

**Parameters**:

* char a: The character to check.

**Steps**:

1. Defines a string operatori containing all valid operators.
2. Iterates through operatori to see if a matches any character.
3. Returns true if a match is found, false otherwise.

**Function: doCalculus**

**Purpose**: Performs the calculations based on the numbers and operators provided.

**Parameters**:

* vector<float> numbers: A vector of numbers to be used in calculations.
* vector<char> operators: A vector of operators that determine which operations to perform.

**Steps**:

1. Initializes a calcs object with the first number.
2. Iterates through the operators vector, applying the corresponding operation to the calcs object.
3. Returns the final value after all operations are applied.

**Main Function (main)**

**Purpose**: The entry point of the program, handling user input and orchestrating the calculation process.

**Steps**:

1. Sets the console title.
2. Enters an infinite loop to continuously prompt the user for input.
3. Prompts the user to type a calculation (or 'h' for help).
4. If the user types 'h', it calls the help function and clears the screen.
5. Parses the input string to separate numbers and operators:
   * If a character is an operator, it adds it to the operators vector and converts the accumulated number string to a float, adding it to the numbers vector.
   * If the character is 'P' (PI) or 'E' (Euler's number), it adds the respective constant to the numbers vector.
6. Converts the final accumulated number string to a float and adds it to the numbers vector.
7. Calls doCalculus with the numbers and operators vectors to compute the result.
8. Prints the result.
9. Pauses and clears the screen before the next iteration.

**Example Workflow**

1. **User Input**: "3+5\*2"
2. **Parsing**:
   * Numbers: [3, 5, 2]
   * Operators: ['+', '\*']
3. **Calculation**:
   * 3 + 5 = 8
   * 8 \* 2 = 16
4. **Output**: "Result: 16"

**Supported Operators**

* + : Addition
* - : Subtraction
* \* : Multiplication
* / : Division
* ^ : Exponentiation
* q : Square root
* Q : Cube root
* c : Cosine
* s : Sine
* t : Tangent
* C : Arccosine
* S : Arcsine
* T : Arctangent
* l : Natural logarithm (base e)
* i : Logarithm base 2
* L : Common logarithm (base 10)
* P : PI
* E : Euler's number