Group 16, LLC

Command Cruncher

User's Manual

Version 1.1

Revision History

Date	Version	Description	Author
01/Dec/24	1.0	Started working on the document.	Emma Roy, Daniel Van Dalsem, Hannah Prosch
08/Dec/24	1.1	Finishing up the document.	Hannah Prosch, Emma Roy, Daniel Van Dalsem, Nifemi Nawal, Warren Tan

Table of Contents

Test Case

1. Purpose

This user manual is included to help the user understand how to properly use the Command Cruncher. Command Cruncher allows the user to compute complex algebraic expressions using basic arithmetic operators, parentheses, and handle operator precedence.

2. Introduction

Command Cruncher is a software program, with the purpose of helping users compute algebraic expressions with ease. Features include arithmetic operators, such as addition, subtraction, multiplication, division, and modulus, as well as considering parentheses.

This program will be run on the command line; thus, the only necessary installations include a code editor (we chose to utilize Visual Studio Code), C/C++ extensions for both compiling and debugging. To install VS Code, one must search up this application on a browser and then download and run the installer. Once the code editor and extensions are installed, click on CompileRun and enter any input to see the desired output. You also have the option of using G++ and make to build it, to do so just have both installed then run make in the folder to build. To run just run the executable.

To install and run Command Cruncher, follow these steps:

1. **Installation:**
   Command Cruncher requires a code editor like Visual Studio Code (VS Code), along with C/C++ extensions for compiling and debugging. To install VS Code:
   - Search for Visual Studio Code in your web browser.
   - Download the installer and run it.
   - After installation, open VS Code and install the C/C++ extension from the marketplace.

2. **Running the Program:**
   - Using Visual Studio Code, open your project folder, then click on CompileRun to compile and run the program.
   - Using G++ and Make: If you prefer a terminal-based approach, make sure you have G++ and make installed. In your terminal, navigate to the project folder, run make to build the executable, and then run the program by typing './(executable_name).'

3. Getting started

To begin, enter the algebraic expression you want Command Cruncher to evaluate. Keep in mind that the supported math operations are addition (+), subtraction (-), multiplication (*), division (/), exponentiation (**), and modulus (%). Command Cruncher also implies multiplication when both terms are presented within parenthesis, such as (3)(4)=12. The results are interpreted by reading the text output displayed after hitting the enter key once your expression has been typed. If you encounter any errors, please reference the troubleshooting section of this document.

4. Advanced features

Currently, Command Cruncher does not include advanced features such as saving/loading defining custom variables, or functions. However, future updates may include these features to enhance functionality.

5. Troubleshooting

1. **Problem: Invalid Expression Format**
   - **Cause:** The expression entered does not follow a valid syntax (e.g., missing operands, unbalanced parentheses).
   - **Solution:** Ensure the expression is correctly formatted and follows arithmetic rules. Command Cruncher will give you the position of the error in the expression, so you can pinpoint where the error originates.
     - Example: (3 + 5) * 2 is valid, while 3 + * 5 is invalid.

2. **Problem: Division by Zero**
   - **Cause:** You attempted to divide by zero.
   - **Solution:** Make sure the denominator in division is not zero.
     - Examples: 5 / 0 will trigger an error.

3. **Problem: Unsupported Operation**
   - **Cause:** The expression uses an operator not supported by the program (e.g., ^ for exponentiation instead of **).
   - **Solution:** Replace ^ with ** for exponentiation.

4. **Problem: Program Crashes on Launch**
   - **Cause:** Missing or incorrect dependencies.
   - **Solution:** Ensure all necessary software is installed properly and dependencies are up to date.

6. Examples

1. **Addition:**
   - Expression: 3 + 5
   - Result: 8

2. **Multiplication:**
   - Expression: 4 * 2
   - Result: 8

3. **Parentheses Handling:**
   - Expression: (6 + 2) * 3
   - Result: 24
   - Explanation: Parentheses ensure 6 + 2 is evaluated first.

4. **Exponentiation:**
   - Expression: 2 ** 3
   - Result: 8

5. **Complex Expression:**
   - Expression: (2 + 3) * (4 – 1) / 2
   - Result: 7.5

6. **Modulus:**
   - Expression: 10 % 3
   - Result: 1

7. **Unary Operators:**
    - Expression: -5 + (+3)
    - Output: -2

7. Glossary of terms

N/A

8. FAQ

1. **Can Command Cruncher handle floating-point numbers?**

- **Answer:** Yes, Command Cruncher supports floating-point numbers in expressions. For example, 2.5 + 3.7 will be correctly evaluated as 6.2.

2. **How do I handle exponentiation in expressions?**

- **Answer:** Use ** for exponentiation. For example, 2 ** 3 will calculate 2 raised to the power of 3, resulting in 8.

3. **Can I include spaces in my expressions?**

- **Answer:** Yes, spaces are ignored during evaluation. For example, 3 + 4 will still be evaluated as 7.

4. **What happens if I divide or use modulus by zero?**

- **Answer:** The program will throw a division by zero or modulus by zero error.

5. **What should I do if I get an "Illegal Character" error?**

- **Answer:** This error occurs when the expression contains supported symbols or characters. Ensure that only supported operators (+, -, *, /, %, **, ()) are used.

6. **How do I stop the program?**

- **Answer:** Type 'stop' when prompted for an expression, and the program will terminate.

7. **Is there a limit to the size of the input expression?**

- **Answer:** Command Cruncher does not explicitly limit the input size, but extremely long expressions may depend on system memory and performance.

8. **What should I do if the program crashes or gives unexpected results?**

- **Answer:** Double-check your input for syntax errors or unsupported features. If the issue persists, ensure all required dependencies are correctly installed and up to date.

9. Can I use custom variables in my expressions?

- **Answer:** No, Command Cruncher does not support variables. It only evaluates numerical expressions.

10. How are errors displayed in the program?

- **Answer:** Errors are displayed in the format:
  - Error: [Error Description]