221005009

B Bukanga

Practical 07 Design

Global Variables:

- inpStr (BYTE array): Stores the user input string.
- RevStr (BYTE array): Stores the reversed string.
- searchChar (BYTE): Stores the character to be searched in the string.
- replaceChar (BYTE): Stores the character that will replace searchChar.

Global Variables

- **inpStr**:
    - **Datatype**: BYTE array (50 bytes)
    - **Description**: This array will hold the input string provided by the user.
- **RevStr**:
    - **Datatype**: BYTE array (50 bytes)
    - **Description**: This array will hold the reversed version of inpStr.
- **searchChar**:
    - **Datatype**: BYTE
    - **Description**: This variable will hold the character that needs to be replaced in inpStr.
- **replaceChar**:
    - **Datatype**: BYTE
    - **Description**: This variable will hold the character that will replace occurrences of searchChar in inpStr.

Functions

a. main proc

- **Purpose**: Entry point of the program. Initializes data, calls functions for string manipulation, and outputs results.
- **Steps**:
    1. **Prompt User**: Display prompts to get the input string from the user.
    2. **Store Input**: Save the user-provided string in inpStr.
    3. **Call Functions**: Call replaceCharacters to perform character replacement.
    4. **Reverse String**: Reverse the modified string.

5. **Output Results**: Display the reversed string and other relevant messages.

b. replaceCharacters proc

- **Purpose**: Replaces occurrences of searchChar in inpStr with replaceChar.
- **Steps**:
    1. **Loop through inpStr**: Check each character of the string.
    2. **Character Comparison**: If a character matches searchChar, replace it with replaceChar.
    3. **Continue**: Move to the next character until the end of the string is reached.

c. reverseString proc

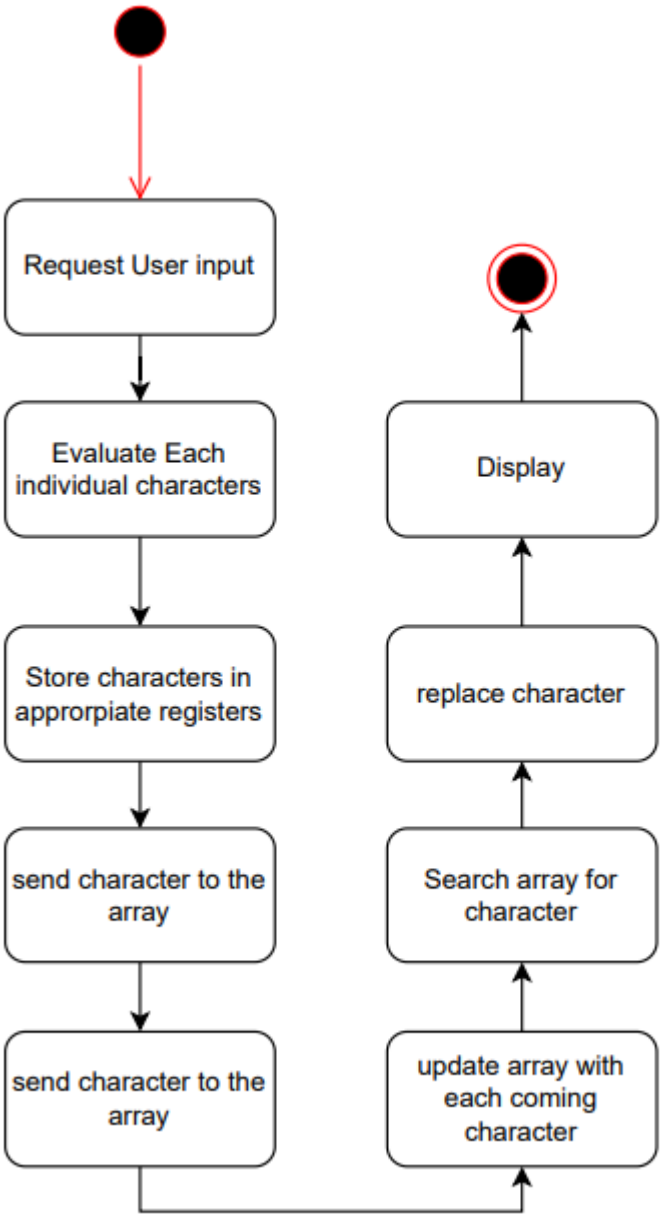
- **Purpose**: Reverses the string stored in inpStr and stores it in RevStr.
- **Steps**:
    1. **Determine Length**: Calculate the length of inpStr.
    2. **Reverse Loop**: Copy characters from the end of inpStr to the beginning of RevStr.
    3. **Null-Terminate**: Ensure RevStr is properly null-terminated.

Algorithm

1. **Initialize Program**:
    - Set up the environment and display initial prompts.
2. **User Input**:
    - Prompt the user to enter a string and store it in inpStr.
3. **Character Replacement**:
    - **Function Call**: replaceCharacters
        - **Input**: inpStr, searchChar, replaceChar
        - **Output**: Modified inpStr with searchChar replaced by replaceChar.
4. **Reverse String**:
    - **Function Call**: reverseString
        - **Input**: Modified inpStr
        - **Output**: Reversed string stored in RevStr.
5. **Display Results**:
    - Output the reversed string stored in RevStr to the user.
6. **Exit Program**:
    - Clean up and terminate the program.

Sample Data Handling

- **Test Data**:
    - **Input**: "simple example$"
    - **searchChar**: '#'
    - **replaceChar**: '!'
    - **Expected Output**: "!elpmaxe elpmis"
- **Test Data**:
    - **Input**: "New%moon"
    - **searchChar**: '%'
    - **replaceChar**: (blank space)
    - **Expected Output**: "noom weN"


Request User input
Evaluate Each individual characters
Store characters in approrpiate registers
send character to the array
send character to the array
update array with each coming character
Search array for character
replace character
Display