Cape Peninsula University of Technology

Faculty of Engineering and the Built Environment
Department of Electrical, Electronic and Computer Engineering

Software Design 1 (SDN150S)

Practical 5
Date of Submission: May 31, 2024.

1. Write a program to calculate the average grade of a student. The grades are stored in an array. Use a function to calculate the average and return it.

   STEPS:

   - Define an array to store the student's grades.
   - Create a function called calculateAverage that takes the grades array and its size as parameters.
   - Inside the function, sum up all the grades, then divide by the number of grades to find the average.
   - Return the average from the function.
   - In the main function, call calculateAverage and print the result.

2. Develop a program that reverses an input string using pointers.

   STEPS:

   - Read a string from the user.
   - Create a function called reverseString that uses a char pointer to reverse the string in place.
   - Use two pointers: one at the start of the string and one at the end.
   - Swap the characters at these pointers, then move them toward the center until they meet or pass each other.
   - Print the reversed string in the main function.

3. Write a C program that asks the user to find all the occurrence of a character, in the string below, while counting the number of occurrence and printing it's position, and pointer address in the string.

   String: "*Future engineering focuses on sustainability, AI integration, renewable energy, advanced materials, and personalized medicine, to enhance efficiency and human well-being*."

   STEPS:

   - Include the necessary header files (**`string.h, ctype.h`**).
   - Declare a character array for the string and a character variable for the target character.

- Prompt the user to input the character.
- Use the **strchr()** function and a loop to iterate through the string, counting occurrences of the character.
- Print the number of occurrence, it's positions in the string, and the pointer address of each position.

4. Create a C program that checks if the entered password meets certain criteria: minimum 8 characters, at least one uppercase letter, and at least one number.

   STEPS:

   - Include the necessary header files (`string.h`).
   - Declare a character array for the password.
   - Prompt the user to enter the password.
   - Use **strstr()** and a loop to check each character and validate against the criteria.
   - Print whether the password is valid or not.

5. Write a C program that takes two strings from the user (first name and last name) and concatenates them with a space between them to display the full name.

   STEPS:

   - Include the necessary header files (`string.h`).
   - Declare character arrays to store the first name, last name, and full name.
   - Prompt the user to input the full name.
   - Use an input function to read the string.
   - Use inbuilt string functions to copy the first name and last name from the full name.
   - Concatenate the first name, a space, and the last name using sprintf() or strcat().
   - Print the full name.