COM2002 INTERMEDIATE PROGRAMMING
2024 – 2025 SPRING

Laboratory Week: 24 – 28 February 2025

Topic : Strings

Program-1 : Dealing a Hand of Cards (deal.c)

Definition : The program deals a random hand from a standard deck of playing cards. Each card in a standard deck has a suit (clubs, diamonds, hearts, or spades) and a rank (two, three, four, five, six, seven, eight, nine, ten, jack, queen, king, or ace). The user will specify how many cards should be in the hand. Then, it prints the full names of the cards.

Expected output:

```
Enter number of the cards in hand: 5
Your hand:
Seven of clubs
Two of spades
Five of diamonds
Ace of spades
Two of hearts
```

Program-2 : Finding smallest and largest words (words.c)

Definition : The program finds the smallest and largest in a series of words. After the user enters the words, the program will determine which words would come first and last if the words were listed in dictionary order. The program must stop accepting input when the user enters a four-letter word. Assume thet no word is more than 20 letters long.

Hint : Use two strings named *smallest_word* and *largest_word* to keep track of the smallest and largest words entered so far. Each time the user enters a new word, use *strcmp* to compare it with *smallest_word*; if the new word is smaller, use *strcpy* to save it in *smallest_word*. Do a similar comparison with *largest_word*. Use *strlen* to determine when the user has entered a four-letter word.

Expected output:

```
Enter word: dog
Enter word: zebra
Enter word: rabbit
Enter word: catfish
Enter word: walrus
Enter word: cat
Enter word: fish

Smallest word: cat
Largest word: zebra
```

Program-3 : Checking Planet Names (planet.c)

Definition : The program is designed to check a series of strings to see which ones are names of planets. The strings are put on the command line.

```
planet.exe Jupiter venus Earth fred
```

The program will indicate whether each string is a planet name and, if it is, display the planet's number.

Expected output:

```
Jupiter is planet 5
venus is not a planet
Earth is planet 3
fred is not a planet
```

Program-4 : Palindrome (palindrome.c)

Definition : The program reads a message and checks whether it is a palindrome. Ignore all characters that are not letters. Use integer variables to keep track of positions in the array.

Palindrome: The letters in the message are the same from left to right as from right to left.

Expected output-1:

```
Enter a message: He lived as a Devil, eh?
Palindrome
```

Expected output-2:

```
Enter a message: How are you?
Not a palindrome
```

Expected output-3:

```
Enter a message: madam
Palindrome
```

Modify the palindrome.c program (palindrome-Modify.c) to use a pointer instead of an integer to keep track of the current position in the array.

Simplify the palindrome-Modify.c program (palindrome-Simplify.c) by taking advantage of the fact that an array name can be used as a pointer.

Modify the palindrome.c program (palindrome-command.c) checks command-line argument whether it is a palindrome.

Program-5 : Printing a One-Month Reminder List (remind.c)

Definition : The program prints a one-month list of daily reminders. The user will enter a series of reminders, with each prefixed by a day of the month. When the user enters 0 instead of a valid day, the program will print a list of all reminders entered, sorted by day.

Overall strategy:

- Read a series of day-and-reminder combinations.
- Store them in order (sorted by day).
- Display them.

scanf will be used to read the days. *read_line* will be used to read the reminders.

The strings will be stored in a two-dimensional array of characters. Each row of the array contains one string. Actions taken after the program reads a day and its associated reminder:

- Search the array to determine where the day belongs, using *strcmp* to do comparisons.
- Use *strcpy* to move all strings below that point down one position.
- Copy the day into the array and call *strcat* to append the reminder to the day.

One complication: how to right-justify the days in a two-character field.

A solution: use *scanf* to read the day into an integer variable, than call *sprintf* to convert the day back into string form.

sprintf is similar to *printf*, except that it writes output into a string.

The call

sprintf(day_str, "%2d", day);

writes the value of day into *day_str*.

The following call of *scanf* ensures that the user doesn't enter more than two digits:

scanf("%2d", &day);

Expected output:

```
Enter day and reminder: 24 Susan's birthday
Enter day and reminder: 5 6:00 - Dinner with Marge and Russ
Enter day and reminder: 26 Movie - "Chinatown"
Enter day and reminder: 7 10:30 - Dental appointment
Enter day and reminder: 12 Movie - "Dazed and Confused"
Enter day and reminder: 5 Saturday class
Enter day and reminder: 12 Saturday class
Enter day and reminder: 0

Day Reminder
 5 Saturday class
 5 6:00 - Dinner with Marge and Russ
 7 10:30 - Dental appointment
12 Saturday class
12 Movie - "Dazed and Confused“
24 Susan's birthday
26 Movie - "Chinatown"
```