COEN 10
Lab 9

Lab 9 - Scheduling System

- Your program will deal with a dentist appointment system.
- The dentist is available each day from 1pm to 7pm for 1-hour appointments.
- Your system creates appointments for one day only.
- The system reserves the earliest appointment available.

Lab 9 – Scheduling System

- Interface
  - The user can use the system to
    - Schedule (1) an appointment
    - Cancel (2) an appointment
    - List (3) the appointments
    - Emergency (4)
    - Count a letter (5)
    - Quit (any other number)

Lab 9 – Scheduling System

- Interface
  - Create an appointment – enter name
    - If there is a free hour
      - The user is asked for his/her name
      - The appointment is scheduled in the earliest free time slot. The hour is shown to the user.
      - Do not allow repetitions
  - Cancel an appointment – enter name
    - If there is an appointment
      - The user is asked for his/her name
      - If found, the appointment is canceled and later appointments are shifted up
  - List appointments
    - Show the schedule, all the names and free slots

Lab 9 – Scheduling System

- Interface
  - Emergency – enter name
    - Schedule an appointment in the first time slot and shift the other ones down
    - Do not allow repetitions
  - Count a letter – enter letter
    - Count the number of occurrences of the letter in all the names in the schedule
  - Quit
    - Return from the main function

Lab 9 – Scheduling System

- Implementation
  - Use an array of strings, size 6x20
    - 6 appointments
  - Initially, the array contains a null character ('\0') in the first position of each string
  - Keep a counter of number of appointments

Lab 9 – Scheduling System

- Implementation
  - Schedule
    - If the dentist is busy, inform the user
    - Otherwise
      - If a reservation exists with that name, tell the user
      - Otherwise
        - An appointment is scheduled in the first slot available, given by the number of appointments (no loop)
        - Update the number of appointments

Lab 9 – Scheduling System

- Implementation
  - Cancellation
    - If the schedule is empty, inform the user
    - Otherwise, search for the name in the array
      - If found
        - Cancel the corresponding appointment
        - Shift later appointments up to close the hole
        - Update the number of appointments
      - If not found
        - tell the user

Lab 9 – Scheduling System

- Implementation
  - List
    - If the schedule is empty, inform the user
    - Otherwise
      - Traverse the array, showing the name assigned to each appointment or an empty string for the free slots.

Lab 9 – Scheduling System

- Implementation
  - Emergency
    - If this is a repetition, inform the user
    - Otherwise
      - If the schedule is full
        - remove the last one and inform the user
      - Schedule the new appointment in the first time slot
      - Shift all the reserved spots down

Lab 9 – Scheduling System

- Implementation
  - Find letters
    - Read the letter
    - Traverse the names, counting the number of occurrences of that letter
    - Use a pointer to traverse each string

Lab 9 – Scheduling System

- Requirements
  - Have a forever loop
    - In the loop, use switch to decide which action to take depending on the command entered: 1, 2, 3, 4, 5 or any other
  - Variables
    - array of strings to keep the appointments
    - number of appointments
  - NEW – emergency and count letter
    - Two more functions

Lab 9 – Scheduling System

- Requirements
  - NEW – count a letter
    - use a pointer to traverse each string when counting the occurrences of the character given
    - Your function will receive the letter (scanf in main) and return the final counter, which is output (printf) in main
    - Declaring function count_letter:

      `int count_letter (char);`

Lab 9 – Scheduling System

- Requirements
  - <u>NEW</u> – count a letter
    - In main, calling count_letter:

```
printf ("letter? ");
__fpurge (stdin);
scanf ("%c", &letter);
__fpurge (stdin);
number = count_letter (letter);
printf ("found %d occurrences of %c\n", number, letter);
```

Lab 9 – Scheduling System

- You will use C in the Mac or Linux
  - Use your DC account
    - The home directory
    - You don't need to do this on the web server
  - Edit the program using vi in the terminal
    - The program needs to be a ".c" file
  - Compile with gcc

    ```
    gcc -o name name.c
    ```
  - Execute

    ```
    ./name
    ```

Lab 9 – Scheduling System

- When you are done
  - Demo
    - Execute your code on the terminal to the TA
  - Submit
    - Print and submit the source code to the TA
    - Don't forget to put your name on it!