BCSL305 Data Structures Lab Experiment - 1

1. a) Develop a Program in C for the following: a) Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String).
   b) Write functions create(), read() and display(); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen.

   Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
   char *name;       // Name of the day
   int date;         // Date of the day
   char *description; // Description of the activity
} Day;

void create(Day *week, int size) {
   for (int i = 0; i < size; i++) {
// Allocate memory for name
     week[i].name = (char *)malloc(20 * sizeof(char));
// Allocate memory for description
      week[i].description = (char *)malloc(100 * sizeof(char));
   }
}

void read(Day *week, int size) {
   for (int i = 0; i < size; i++) {
```

```
        printf("Enter name of day %d: ", i + 1);
        scanf("%s", week[i].name);
        printf("Enter date: ");
        scanf("%d", &week[i].date);
        printf("Enter description: ");
        scanf(" %[^\n]", week[i].description); // Read a line of text including spaces
    }
}

void display(Day *week, int size) {
    printf("\nWeek's Activity Details:\n");
    for (int i = 0; i < size; i++) {
        printf("Day: %s, Date: %d, Activity: %s\n", week[i].name, week[i].date, week[i].description);
    }
}

int main() {
    int size = 7; // 7 days in a week
    Day *week = (Day *)malloc(size * sizeof(Day)); // Dynamically allocate the array

    create(week, size); // Allocate memory for names and descriptions
    read(week, size);   // Read data from user
    display(week, size); // Display the week's activities

    // Free the allocated memory
    for (int i = 0; i < size; i++) {
        free(week[i].name);
        free(week[i].description);
    }
    free(week);
    return 0;
}
```