

1. Develop a Program in C for the following:

- 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).
- 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.

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

struct Day {
    char *dayName;
    int date;
    char *activity;
};

void create(struct Day *day) {
    day->dayName = (char *)malloc(sizeof(char) * 20);
    day->activity = (char *)malloc(sizeof(char) * 100);

    printf("Enter the day name: ");
    scanf("%s", day->dayName);

    printf("Enter the date: ");
    scanf("%d", &day->date);

    printf("Enter the activity for the day: ");
    scanf(" %[^\\n]s", day->activity);
}

void read(struct Day *calendar, int size) {
    for (int i = 0; i < size; i++) {
        printf("Enter details for Day %d:\\n", i + 1);
        create(&calendar[i]);
    }
}

void display(struct Day *calendar, int size) {
    printf("\\nWeek's Activity Details:\\n");
    for (int i = 0; i < size; i++) {
        printf("Day %d:\\n", i + 1);
        printf("Day Name: %s\\n", calendar[i].dayName);
        printf("Date: %d\\n", calendar[i].date);
        printf("Activity: %s\\n", calendar[i].activity);
        printf("\\n");
    }
}

void freeMemory(struct Day *calendar, int size) {
    for (int i = 0; i < size; i++) {
```

```

    free(calendar[i].dayName);
    free(calendar[i].activity);
}
}

int main() {
    int size;
    printf("Enter the number of days in the week: ");
    scanf("%d", &size);

    struct Day *calendar = (struct Day *)malloc(sizeof(struct Day) * size);

    if (calendar == NULL) {
        printf("Memory allocation failed. Exiting program.\n");
        return 1;
    }

    read(calendar, size);
    display(calendar, size);

    freeMemory(calendar, size);
    free(calendar);

    return 0;
}

```

OUTPUT:

```

Enter the number of days in the week: 7

Enter details for Day 1:
Enter the day name: Sunday
Enter the date: 1
Enter the activity for the day: Learning

Enter details for Day 2:
Enter the day name: Monday
Enter the date: 2
Enter the activity for the day: Coding

Enter details for Day 3:
Enter the day name: Tuesday
Enter the date: 3
Enter the activity for the day: Testing

Enter details for Day 4:
Enter the day name: Wednesday
Enter the date: 4
Enter the activity for the day: Debugging

Enter details for Day 5:
Enter the day name: Thursday
Enter the date: 5
Enter the activity for the day: Publishing

Enter details for Day 6:
Enter the day name: Friday
Enter the date: 6
Enter the activity for the day: Marketing

```

Enter details *for* Day 7:
Enter the day name: Saturday
Enter the date: 7
Enter the activity *for* the day: Earning

Week's Activity Details:

Day 1:
Day Name: Sunday
Date: 1
Activity: Learning

Day 2:
Day Name:
Monday
Date: 2
Activity: Coding

Day 3:
Day Name:
Tuesday
Date: 3
Activity: Testing

Day 4:
Day Name:
Wednesday
Date: 4
Activity: Debugging

Day 5:
Day Name:
Thursday
Date: 5
Activity: Publishing

Day 6:
Day Name:
Friday
Date: 6
Activity: Marketing

Day 7:
Day Name:
Saturday
Date: 7
Activity: Earning