Calendar

```
#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;
}
```