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

// Structure to represent a day in the calendar
struct Day {
    char *name;
    int date;
    char *activity;
};

// Function to create a day and populate its fields
struct Day createDay() {
    struct Day day;

    day.name = (char *)malloc(20 * sizeof(char)); // Allocating
memory for the name
    printf("Enter the name of the day: ");
    scanf("%s", day.name);

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

    day.activity = (char *)malloc(100 * sizeof(char)); //
Allocating memory for the activity description
    printf("Enter the activity for the day: ");
    scanf(" %[^\n]s", day.activity);

    return day;
}

// Function to check for duplicate entries
int isDuplicate(struct Day calendar[7], struct Day newDay, int
numDays) {
    for (int i = 0; i < numDays; i++) {
        if (strcmp(calendar[i].name, newDay.name) == 0) {
            return 1; // Duplicate found
        }
    }
    return 0; // No duplicate
}

// Function to read data for the calendar from the keyboard
void read(struct Day calendar[7]) {
    int numDays = 0;
    while (numDays < 7) {
        printf("\nEnter details for day %d:\n", numDays + 1);
        struct Day newDay = createDay();

        if (!isDuplicate(calendar, newDay, numDays)) {
            calendar[numDays] = newDay;
            numDays++;
        } else {
            printf("Duplicate entry. Please enter a different
day.\n");
            free(newDay.name);
            free(newDay.activity);
        }
    }
}

// Function to display the calendar
void display(struct Day calendar[7]) {
    printf("\nCalendar for the week:\n");
    for (int i = 0; i < 7; i++) {
        printf("Day %d: %s, Date: %d\n", i + 1,
calendar[i].name, calendar[i].date);
        printf("Activity: %s\n\n", calendar[i].activity);
    }
}

// Function to free dynamically allocated memory
void freeMemory(struct Day calendar[7]) {
    for (int i = 0; i < 7; i++) {
        free(calendar[i].name);
        free(calendar[i].activity);
    }
}

int main() {
    struct Day calendar[7];

    printf("Create a weekly calendar:\n");
    read(calendar);

    display(calendar);

    // Free allocated memory
    freeMemory(calendar);
    return 0;
}
```