1. 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.

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

struct day{
    char *dayname;
    int d,m,y;
    char *activitydescription;
};

void create(struct day *calendar){
        char *daynames[]= \
                {"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"};
        for(int i=0;i<7;i++){
                calendar[i].dayname=strdup(daynames[i]);
                size_t bufferSize=256;
                calendar[i].activitydescription=(char *)malloc(bufferSize*sizeof(char));
        }
}

void read(struct day *calendar){
        for(int i=0;i<7;i++){
                printf("Enter date for %s in dd/mm/yy : ",calendar[i].dayname);
                scanf("%d%d%d",&calendar[i].d,&calendar[i].m,&calendar[i].y);
                printf("Enter activity for %s : ",calendar[i].dayname);
                while(getchar()!='\n')
                        ;
                size_t bufferSize=256;
                getline(&calendar[i].activitydescription,&bufferSize,stdin);
        }
}

void display(struct day *calendar){
        printf("%-10s %-10s %-10s\n","Day","Date","Activity");
        for(int i=0;i<7;++i){
                printf("%-10s %d/%d/%d %-10s\n", \
                        calendar[i].dayname,calendar[i].d, \
                        calendar[i].m,calendar[i].y, \
                        calendar[i].activitydescription
                );
        }
}

int main(){
        struct day *calendar=(struct day *)malloc(7*sizeof(struct day));
        if(calendar==NULL){
                fprintf(stderr,"Memory allocation failed\n");
                return 1;
        }

        create(calendar);
        read(calendar);
        display(calendar);

        for(int i=0;i<7;++i){
                free(calendar[i].dayname);
                free(calendar[i].activitydescription);
        }
        free(calendar);
        return 0;
}
```

Tips and Tricks

Structure :
day
Functions :

1. void create

2. void read

3. display

4. main