Advanced C Programming - Assignment 1

Name	T K Gowtham
Email ID	gowthamkamalasekar@gmail.com
College	VIT Chennai

Topic 1: Structures

C program that represents a **calendar for a week**. Each day has:

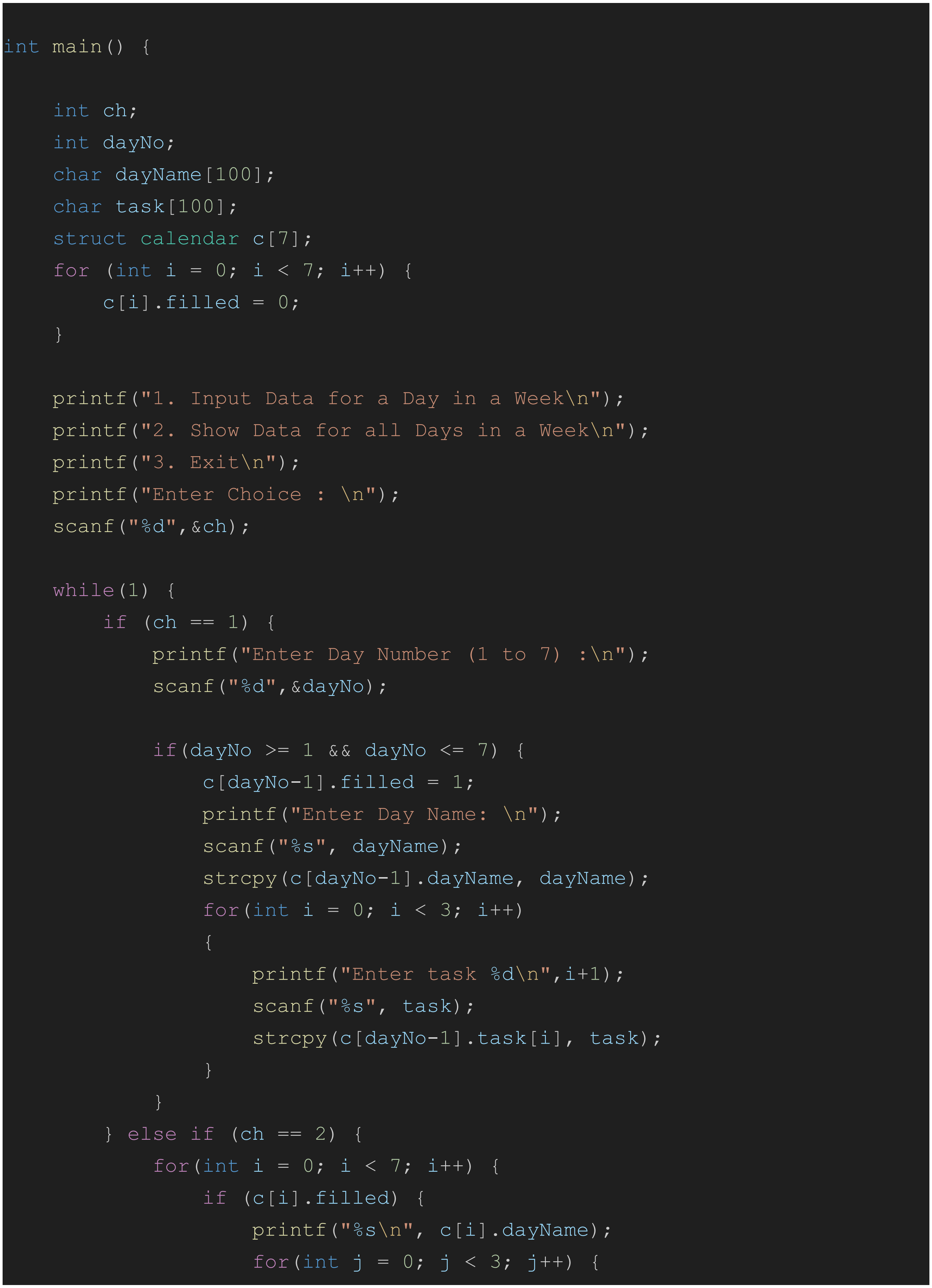
- dayName (e.g., "Monday")
- tasks (array of strings with maximum 3 tasks per day)

Note:

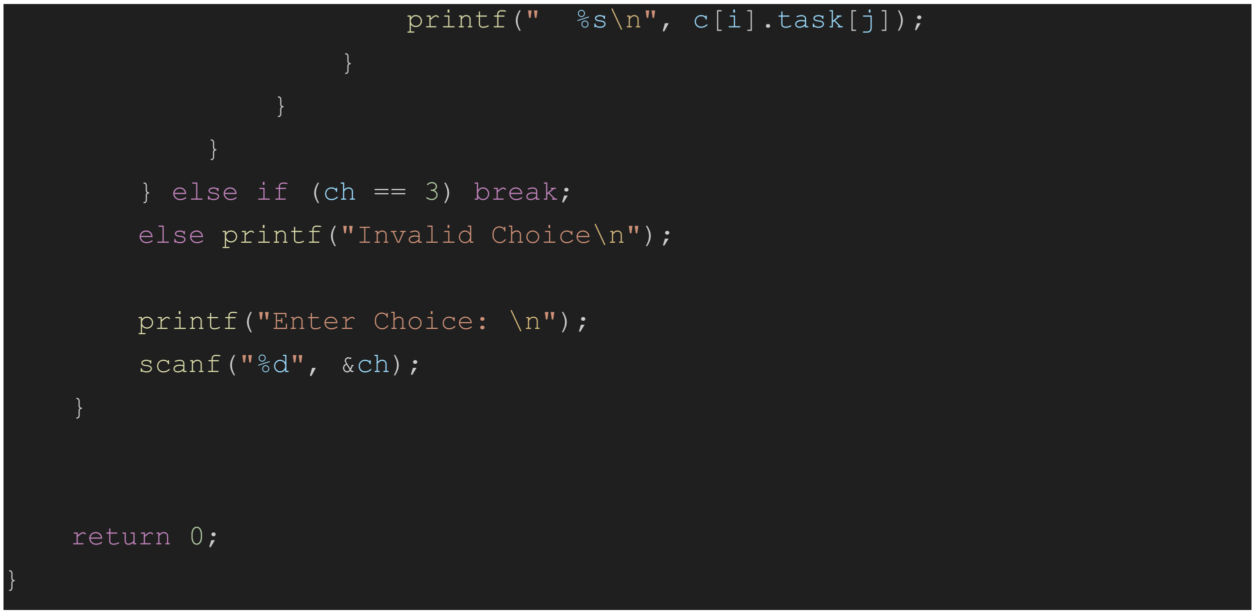
1. Define appropriate structures.
2. Allow the user to input tasks for any day.
3. Display all tasks grouped by the day.

Code :

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

struct calendar
{
    char dayName[100];
    char task[3][100];
    int filled;
};
```

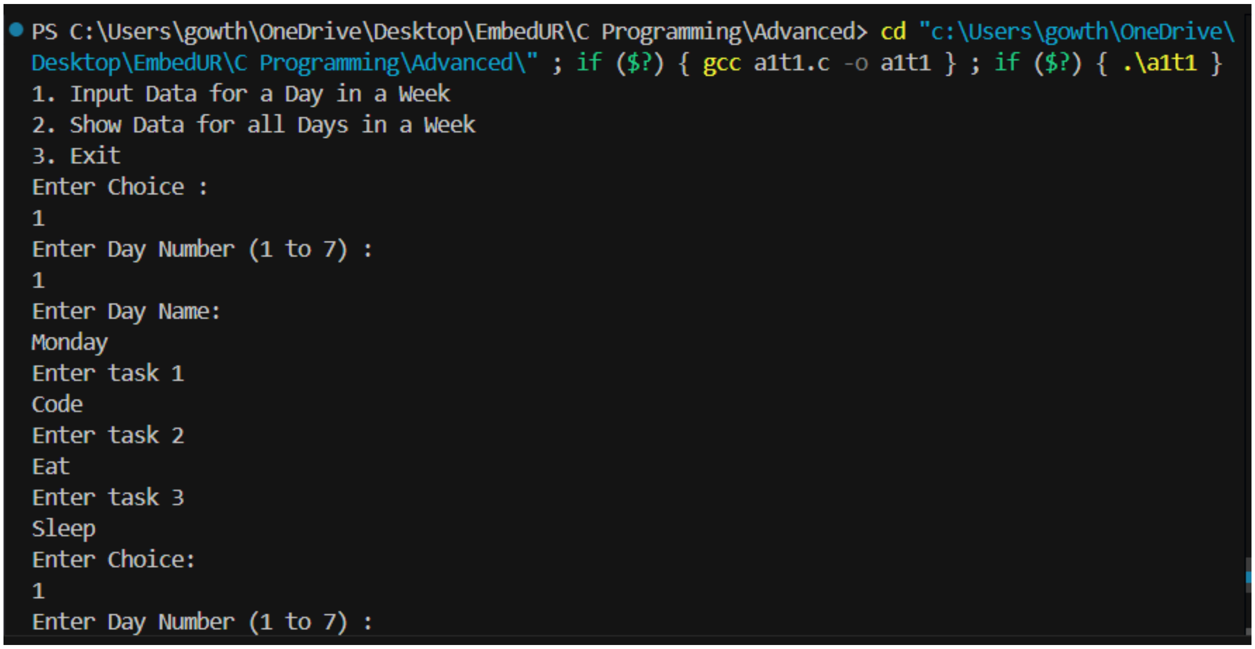


```
int main() {

    int ch;
    int dayNo;
    char dayName[100];
    char task[100];
    struct calendar c[7];
    for (int i = 0; i < 7; i++) {
        c[i].filled = 0;
    }

    printf("1. Input Data for a Day in a Week\n");
    printf("2. Show Data for all Days in a Week\n");
    printf("3. Exit\n");
    printf("Enter Choice : \n");
    scanf("%d",&ch);

    while(1) {
        if (ch == 1) {
            printf("Enter Day Number (1 to 7) :\n");
            scanf("%d",&dayNo);

            if(dayNo >= 1 && dayNo <= 7) {
                c[dayNo-1].filled = 1;
                printf("Enter Day Name: \n");
                scanf("%s", dayName);
                strcpy(c[dayNo-1].dayName, dayName);
                for(int i = 0; i < 3; i++)
                {
                    printf("Enter task %d\n",i+1);
                    scanf("%s", task);
                    strcpy(c[dayNo-1].task[i], task);
                }
            }
        } else if (ch == 2) {
            for(int i = 0; i < 7; i++) {
                if (c[i].filled) {
                    printf("%s\n", c[i].dayName);
                    for(int j = 0; j < 3; j++) {
```

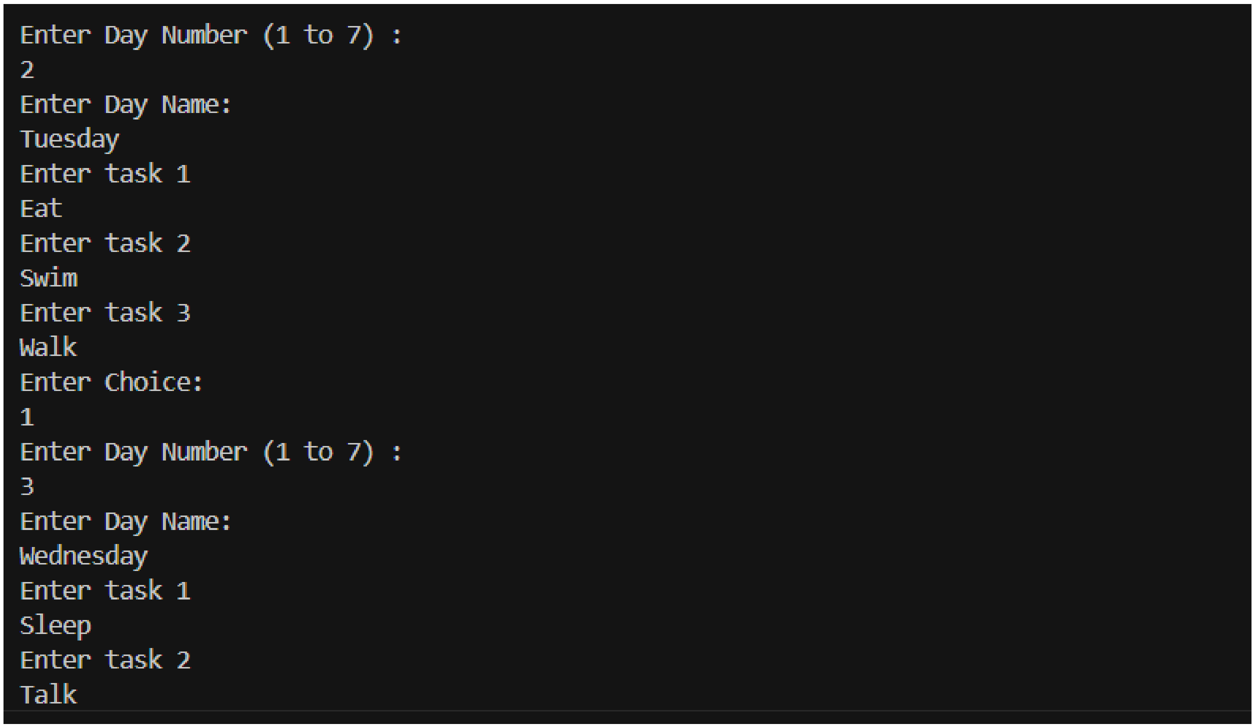


```c
                        printf("  %s\n", c[i].task[j]);
                    }
                }
            }
        } else if (ch == 3) break;
        else printf("Invalid Choice\n");

        printf("Enter Choice: \n");
        scanf("%d", &ch);
    }


    return 0;
}
```

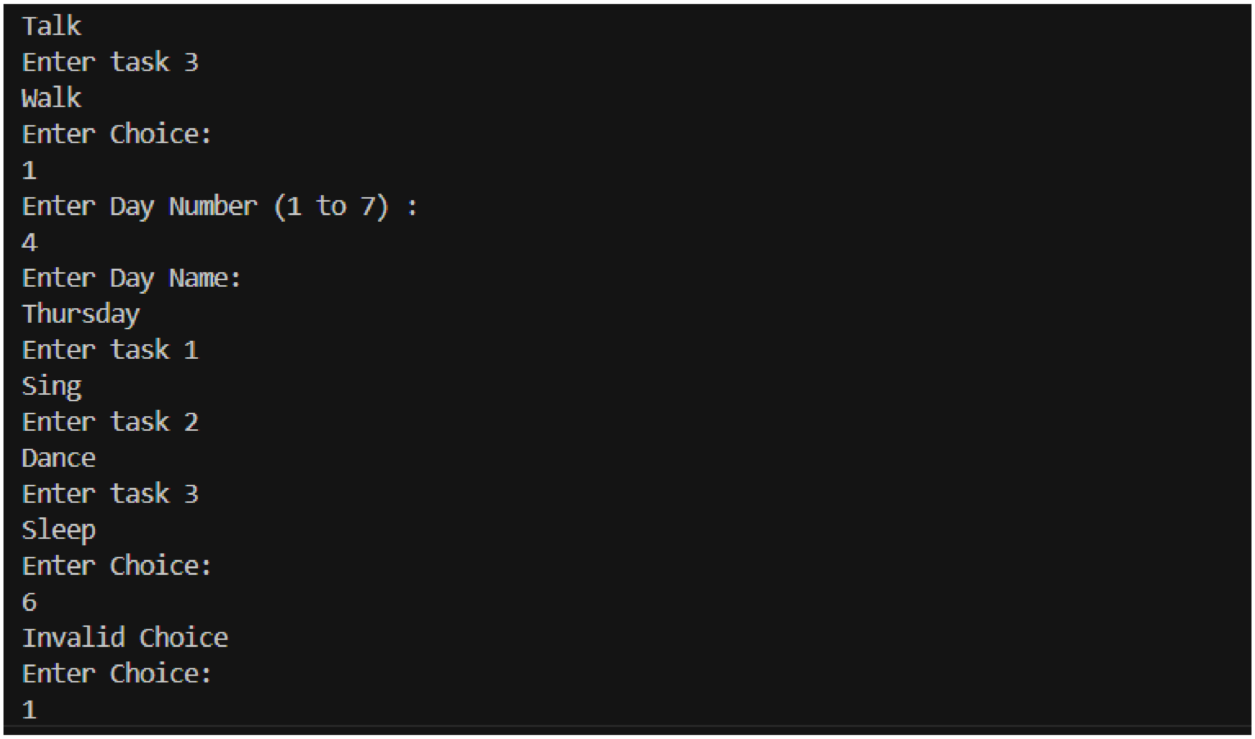
Output :



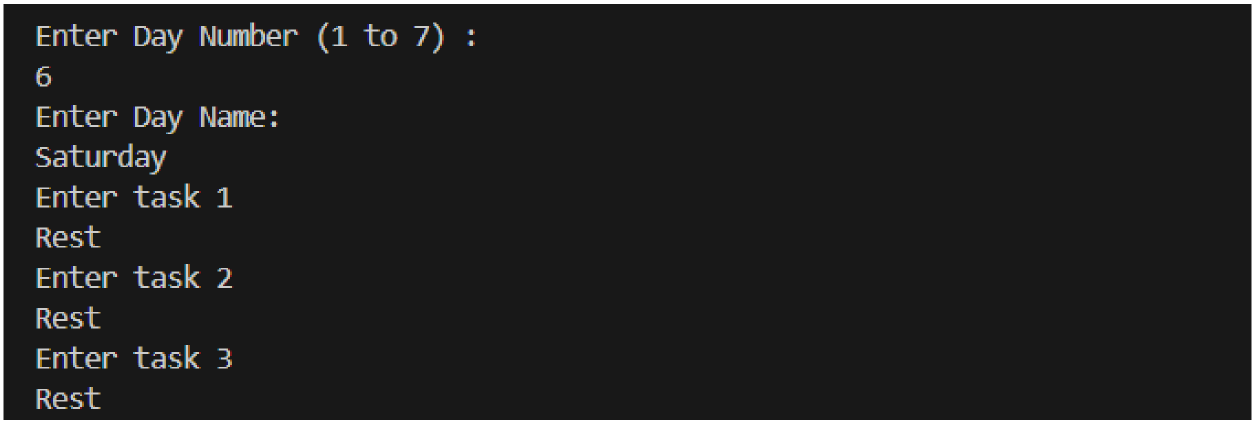
```
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced> cd "c:\Users\gowth\OneDrive\
Desktop\EmbedUR\C Programming\Advanced\" ; if ($?) { gcc a1t1.c -o a1t1 } ; if ($?) { .\a1t1 }
1. Input Data for a Day in a Week
2. Show Data for all Days in a Week
3. Exit
Enter Choice :
1
Enter Day Number (1 to 7) :
1
Enter Day Name:
Monday
Enter task 1
Code
Enter task 2
Eat
Enter task 3
Sleep
Enter Choice:
1
Enter Day Number (1 to 7) :
```



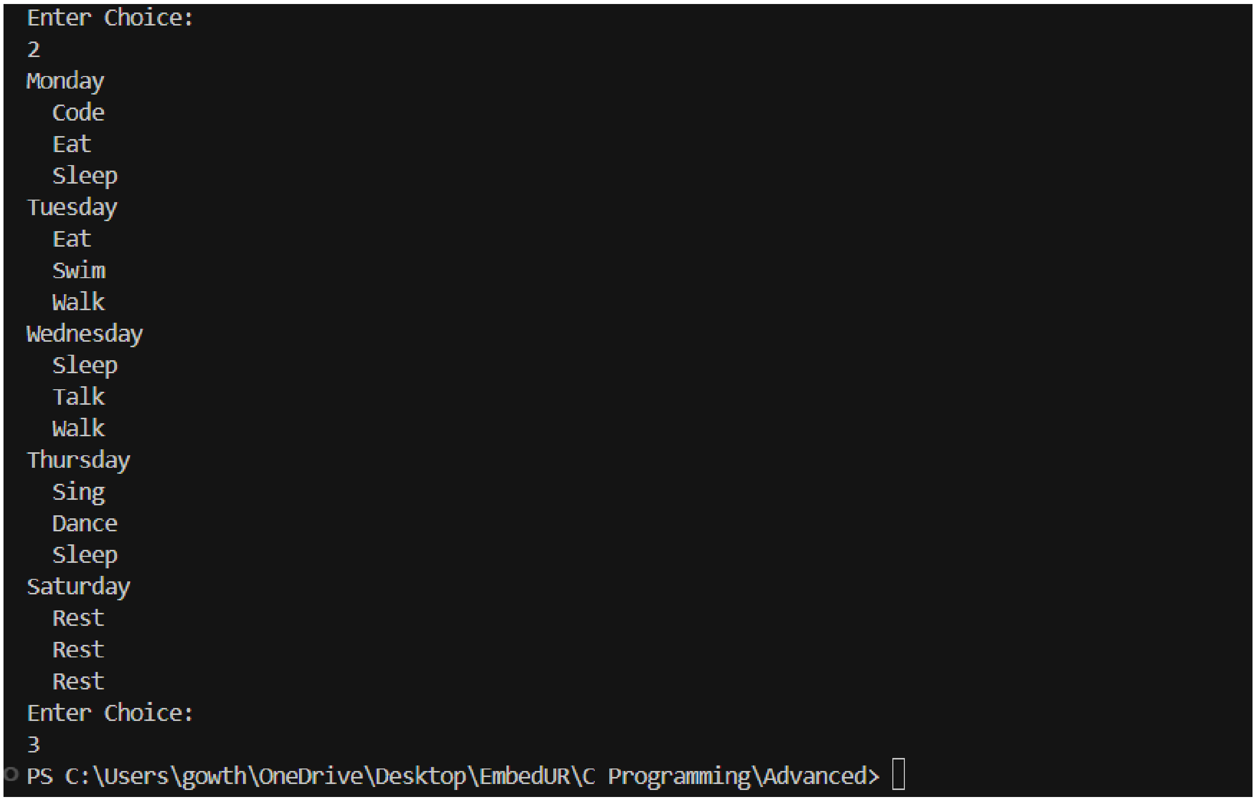
```
Enter Day Number (1 to 7) :
2
Enter Day Name:
Tuesday
Enter task 1
Eat
Enter task 2
Swim
Enter task 3
Walk
Enter Choice:
1
Enter Day Number (1 to 7) :
3
Enter Day Name:
Wednesday
Enter task 1
Sleep
Enter task 2
Talk
```



```
Talk
Enter task 3
Walk
Enter Choice:
1
Enter Day Number (1 to 7) :
4
Enter Day Name:
Thursday
Enter task 1
Sing
Enter task 2
Dance
Enter task 3
Sleep
Enter Choice:
6
Invalid Choice
Enter Choice:
1
```



```
Enter Day Number (1 to 7) :
6
Enter Day Name:
Saturday
Enter task 1
Rest
Enter task 2
Rest
Enter task 3
Rest
```



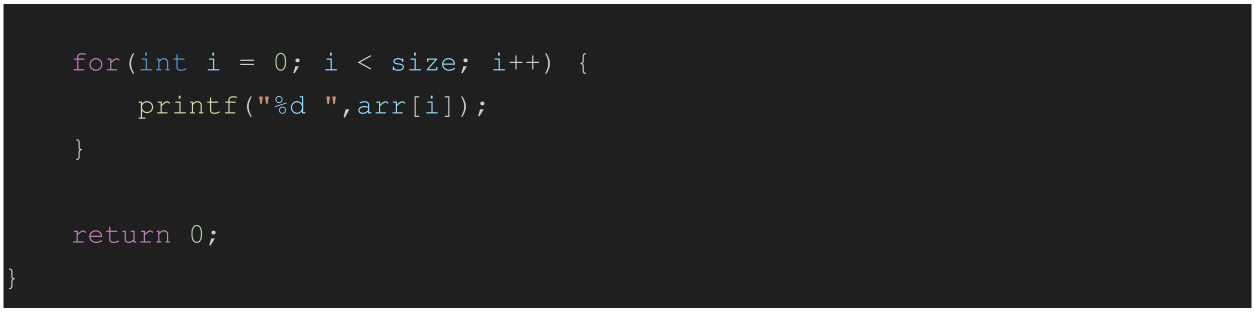
```
Enter Choice:
2
Monday
  Code
  Eat
  Sleep
Tuesday
  Eat
  Swim
  Walk
Wednesday
  Sleep
  Talk
  Walk
Thursday
  Sing
  Dance
  Sleep
Saturday
  Rest
  Rest
  Rest
Enter Choice:
3
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced>
```

Topic 2: Pointers

Question:

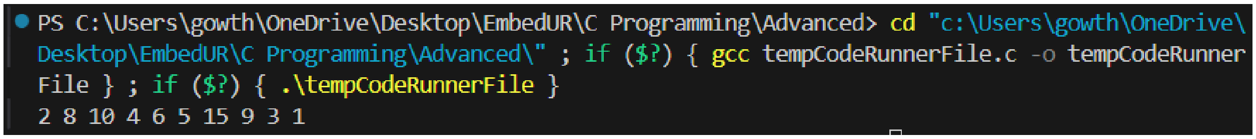
Write a function in C that takes a pointer to an integer array and its size, and then rearranges the array in-place such that all even numbers appear before odd numbers, preserving the original relative order using only pointer arithmetic (no indexing with []).

```c
#include <stdio.h>

void rearrage(int *arr, int size) {
    int even[10];
    int odd[10];
    int osize = 0;
    int esize = 0;

    for(int i = 0; i < size; i++){
        if (*(arr + i) % 2 == 0) {
           *(even + esize) = *(arr + i);
           esize++;
        } else {
            *(odd + osize) = *(arr + i);
            osize++;
        }
    }

    int* ptr = arr;
    for(int i = 0; i < esize; i++){
        *(ptr++) = *(even + i);
    }

    for(int i = 0; i < osize; i++) {
        *(ptr++) = *(odd + i);
    }
}

int main() {
    int size = 10;
    int arr[] = {5, 15, 2, 8, 10, 9, 4, 3, 1, 6};

    rearrage(arr, size);
```



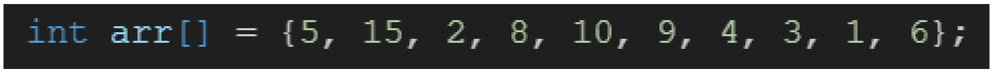
```c
    for(int i = 0; i < size; i++) {
        printf("%d ",arr[i]);
    }

    return 0;
}
```

Output :



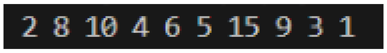
```
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced> cd "c:\Users\gowth\OneDrive\
Desktop\EmbedUR\C Programming\Advanced\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunner
File } ; if ($?) { .\tempCodeRunnerFile }
2 8 10 4 6 5 15 9 3 1
```

Original Array :



```c
int arr[] = {5, 15, 2, 8, 10, 9, 4, 3, 1, 6};
```

Modified Array :



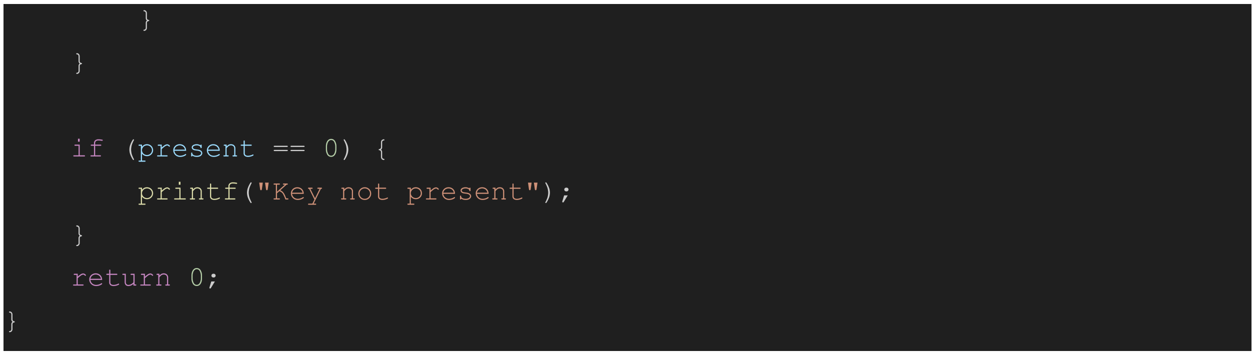
```
2 8 10 4 6 5 15 9 3 1
```

Topic 3: Arrays

Question:
You are given a 2D matrix of size n x n where each row and each column is sorted in increasing order. Write a C function to determine whether a given key exists in the matrix using the most efficient approach.

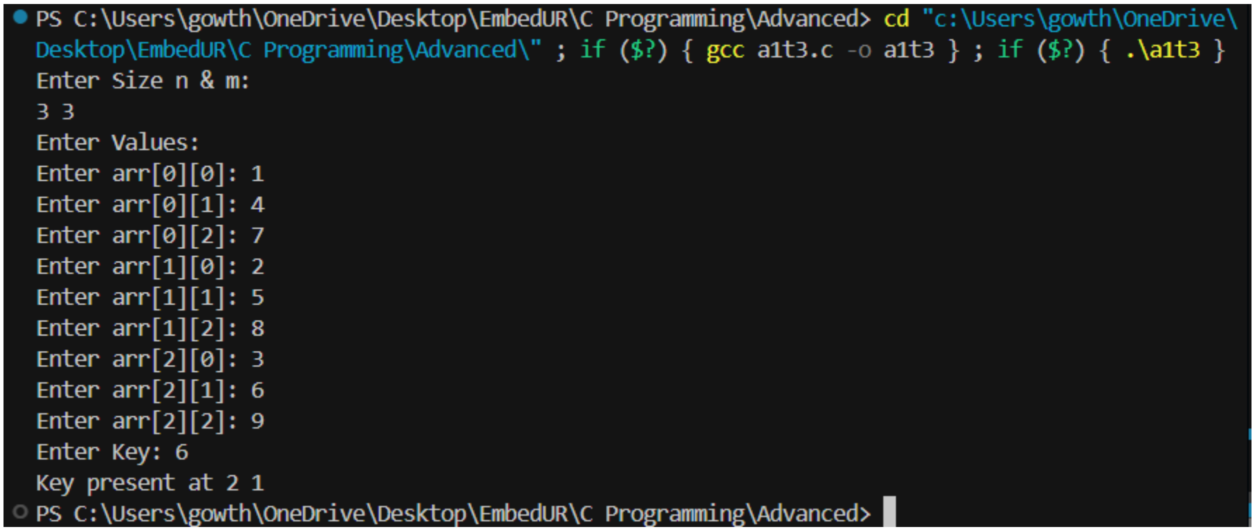
Code :

```c
#include <stdio.h>

int main() {
    int n, m;
    printf("Enter Size n & m: \n");
    scanf("%d %d",&n,&m);

    int arr[n][m];

    printf("Enter Values:  \n");
    for(int i = 0;i < n;i++){
        for(int j = 0; j < n; j++){
            printf("Enter arr[%d][%d]: ",i,j);
            scanf("%d",&arr[i][j]);
        }
    }

    printf("Enter Key: ");
    int key;
    scanf("%d",&key);

    int row = 0; int col = m - 1;
    int present = 0;
    while(row < m && col >= 0) {
        if(arr[row][col] == key) {
            printf("Key present at %d %d\n",row,col);
            present = 1;
            break;
        } else if (arr[row][col] > key ) {
            col--;
        } else {
            row++;
```

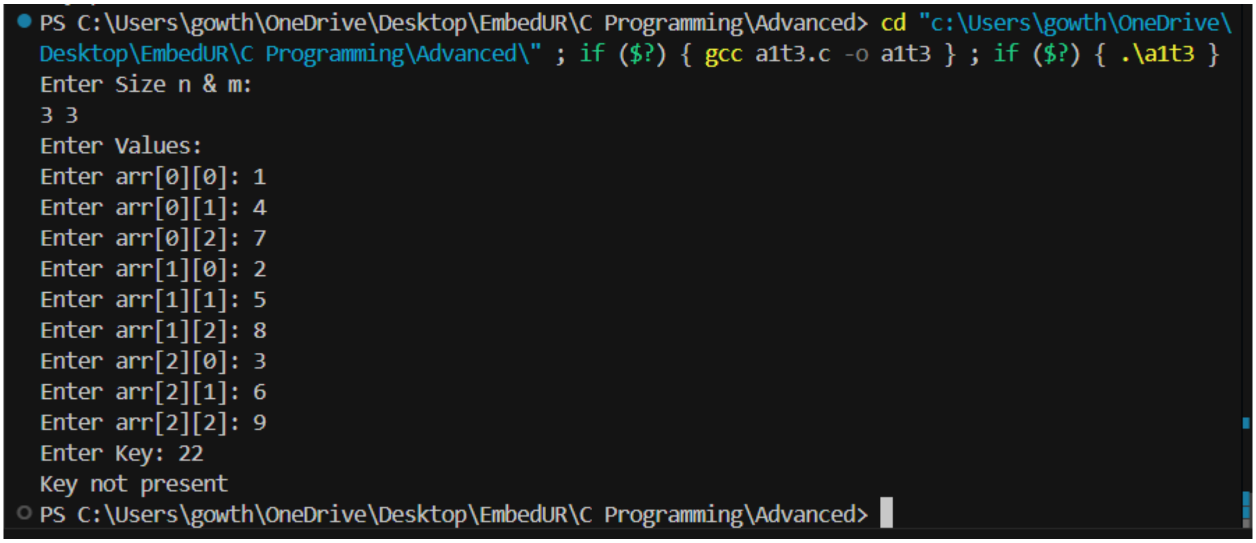


```
        }
    }

    if (present == 0) {
        printf("Key not present");
    }
    return 0;
}
```

Output :



```
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced> cd "c:\Users\gowth\OneDrive\
Desktop\EmbedUR\C Programming\Advanced\" ; if ($?) { gcc a1t3.c -o a1t3 } ; if ($?) { .\a1t3 }
Enter Size n & m:
3 3
Enter Values:
Enter arr[0][0]: 1
Enter arr[0][1]: 4
Enter arr[0][2]: 7
Enter arr[1][0]: 2
Enter arr[1][1]: 5
Enter arr[1][2]: 8
Enter arr[2][0]: 3
Enter arr[2][1]: 6
Enter arr[2][2]: 9
Enter Key: 6
Key present at 2 1
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced>
```



```
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced> cd "c:\Users\gowth\OneDrive\
Desktop\EmbedUR\C Programming\Advanced\" ; if ($?) { gcc a1t3.c -o a1t3 } ; if ($?) { .\a1t3 }
Enter Size n & m:
3 3
Enter Values:
Enter arr[0][0]: 1
Enter arr[0][1]: 4
Enter arr[0][2]: 7
Enter arr[1][0]: 2
Enter arr[1][1]: 5
Enter arr[1][2]: 8
Enter arr[2][0]: 3
Enter arr[2][1]: 6
Enter arr[2][2]: 9
Enter Key: 22
Key not present
PS C:\Users\gowth\OneDrive\Desktop\EmbedUR\C Programming\Advanced>
```