

Module 1 Assignment

Topic 1: Structures – Calendar for a Week

Code:

```
#include <stdio.h>

#include <string.h>


#define MAX_TASKS 3

#define MAX_LEN 50

#define DAYS_IN_WEEK 7


typedef struct {

    char dayName[10];

    char tasks[MAX_TASKS][MAX_LEN];

    int taskCount;

} Day;


void inputTasks(Day *week, int index) {

    printf("Enter number of tasks for %s (max 3): ", week[index].dayName);

    scanf("%d", &week[index].taskCount);

    getchar(); // consume newline


    for (int i = 0; i < week[index].taskCount; ++i) {

        printf("Enter task %d: ", i + 1);

        fgets(week[index].tasks[i], MAX_LEN, stdin);

        week[index].tasks[i][strcspn(week[index].tasks[i], "\n")] = 0; // remove newline

    }

}
```

```

void displayTasks(Day *week) {
    for (int i = 0; i < DAYS_IN_WEEK; ++i) {
        printf("\n%s:\n", week[i].dayName);
        for (int j = 0; j < week[i].taskCount; ++j) {
            printf("  - %s\n", week[i].tasks[j]);
        }
    }
}

int main() {
    Day week[DAYS_IN_WEEK] = {
        {"Monday"}, {"Tuesday"}, {"Wednesday"}, {"Thursday"},
        {"Friday"}, {"Saturday"}, {"Sunday"}
    };
    int index;
    char choice;
    do {
        printf("\nEnter day index to input tasks (0 = Monday, 6 = Sunday): ");
        scanf("%d", &index);
        if (index >= 0 && index < DAYS_IN_WEEK) {
            inputTasks(week, index);
        }
        printf("Do you want to enter tasks for another day? (y/n): ");
        scanf(" %c", &choice);
    } while (choice == 'y');

    displayTasks(week);
    return 0;}

```

Code Snippet:

```
main.c
1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAX_TASKS 3
5 #define MAX_LEN 50
6 #define DAYS_IN_WEEK 7
7
8 typedef struct {
9     char dayName[10];
10    char tasks[MAX_TASKS][MAX_LEN];
11    int taskCount;
12 } Day;
13
14 void inputTasks(Day *week, int index) {
15     printf("Enter number of tasks for %s (max 3): ", week[index].dayName);
16     scanf("%d", &week[index].taskCount);
17     getchar(); // consume newline
18
19     for (int i = 0; i < week[index].taskCount; ++i) {
20         printf("Enter task %d: ", i + 1);
21         fgets(week[index].tasks[i], MAX_LEN, stdin);
22         week[index].tasks[i][strcspn(week[index].tasks[i], "\n")] = 0; // remove newline
23     }
24 }
25
26 void displayTasks(Day *week) {
27     for (int i = 0; i < DAYS_IN_WEEK; ++i) {
28         printf("\n%s:\n", week[i].dayName);
29         for (int j = 0; j < week[i].taskCount; ++j) {
30             printf("  - %s\n", week[i].tasks[j]);
31         }
32     }
33 }
34
35 int main() {
36     Day week[DAYS_IN_WEEK] = {
37         {"Monday"}, {"Tuesday"}, {"Wednesday"}, {"Thursday"},
38         {"Friday"}, {"Saturday"}, {"Sunday"}
39     };
40
41     int index;
42     char choice;
43
44     do {
45         printf("\nEnter day index to input tasks (0 = Monday, 6 = Sunday): ");
46         scanf("%d", &index);
47         if (index >= 0 && index < DAYS_IN_WEEK) {
48             inputTasks(week, index);
49         }
50         printf("Do you want to enter tasks for another day? (y/n): ");
51         scanf(" %c", &choice);
52     } while (choice == 'y');
53
54     displayTasks(week);
55     return 0;
56 }
57
```

Output Snippet:

```
Enter day index to input tasks (0 = Monday, 6 = Sunday): 1
Enter number of tasks for Tuesday (max 3): 2
Enter task 1: Report submission
Enter task 2: car Delivery
Do you want to enter tasks for another day? (y/n): y

Enter day index to input tasks (0 = Monday, 6 = Sunday): 5
Enter number of tasks for Saturday (max 3): 1
Enter task 1: temple
Do you want to enter tasks for another day? (y/n): n

Monday:

Tuesday:
  - Report submission
  - car Delivery

Wednesday:

Thursday:

Friday:

Saturday:
  - temple

Sunday:
```

Topic 2: Pointers – Rearranging Array with Only Pointers

Code:

```
#include <stdio.h>

void rearrangeEvenOdd(int *arr, int size) {
    int temp[size];
    int *evenPtr = temp;
    int *oddPtr = temp;
    for (int i = 0; i < size; ++i) {
        if (*(arr + i) % 2 == 0)
            ++oddPtr;
    }
    evenPtr = temp;
    int *startOdd = oddPtr;
    for (int i = 0; i < size; ++i) {
        if (*(arr + i) % 2 == 0)
            *(evenPtr++) = *(arr + i);
        else
            *(oddPtr++) = *(arr + i);}
    for (int i = 0; i < size; ++i)
        *(arr + i) = *(temp + i);}

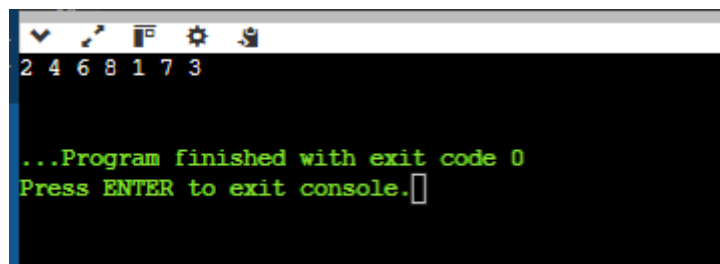
void printArray(int *arr, int size) {
    for (int i = 0; i < size; ++i)
        printf("%d ", *(arr + i));
    printf("\n");}

int main() {
    int arr[] = {1, 2, 4, 7, 6, 3, 8};
    int size = sizeof(arr) / sizeof(arr[0]);
    rearrangeEvenOdd(arr, size);
    printArray(arr, size);
    return 0;}
```

Code Snippet:

```
main.c
1  #include <stdio.h>
2  void rearrangeEvenOdd(int *arr, int size) {
3      int temp[size];
4      int *evenPtr = temp;
5      int *oddPtr = temp;
6
7      // Move oddPtr to the end of the even section
8      for (int i = 0; i < size; ++i) {
9          if (*(arr + i) % 2 == 0)
10             ++oddPtr;
11     }
12
13     evenPtr = temp;
14     int *startOdd = oddPtr;
15
16     for (int i = 0; i < size; ++i) {
17         if (*(arr + i) % 2 == 0)
18             *(evenPtr++) = *(arr + i);
19         else
20             *(oddPtr++) = *(arr + i);
21     }
22     // Copy back
23     for (int i = 0; i < size; ++i)
24         *(arr + i) = *(temp + i);
25 }
26 void printArray(int *arr, int size) {
27     for (int i = 0; i < size; ++i)
28         printf("%d ", *(arr + i));
29     printf("\n");
30 }
31 int main() {
32     int arr[] = {1, 2, 4, 7, 6, 3, 8};
33     int size = sizeof(arr) / sizeof(arr[0]);
34     rearrangeEvenOdd(arr, size);
35     printArray(arr, size);
36     return 0;
37 }
38
```

Output Snippet:



```
2 4 6 8 1 7 3

...Program finished with exit code 0
Press ENTER to exit console.
```

Topic 3: Arrays – Search in Sorted 2D Matrix

Code:

```
#include <stdio.h>

int searchMatrix(int n, int matrix[n][n], int key) {
    int row = 0, col = n - 1;
    while (row < n && col >= 0) {
        if (matrix[row][col] == key)
            return 1;
        else if (matrix[row][col] > key)
            col--;
        else
            row++;
    }
    return 0;
}

int main() {
    int matrix[4][4] = {
        {1, 4, 7, 11},
        {2, 5, 8, 12},
        {3, 6, 9, 16},
        {10, 13, 14, 17}
    };
    int key = 14;
    if (searchMatrix(4, matrix, key))
        printf("Key found\n");
    else
        printf("Key not found\n");
    return 0;
}
```

Code Snippet:

```
main.c
1  #include <stdio.h>
2  int searchMatrix(int n, int matrix[n][n], int key) {
3      int row = 0, col = n - 1;
4      while (row < n && col >= 0) {
5          if (matrix[row][col] == key)
6              return 1;
7          else if (matrix[row][col] > key)
8              col--;
9          else
10             row++;
11     }
12     return 0;
13 }
14 int main() {
15     int matrix[4][4] = {
16         {1, 4, 7, 11},
17         {2, 5, 8, 12},
18         {3, 6, 9, 16},
19         {10, 13, 14, 17}
20     };
21     int choice = 1;
22     while(choice){
23         int key;
24         printf("Enter the key to find : ");
25         scanf("%d",&key);
26         if (searchMatrix(4, matrix, key))
27             printf("Key found\n");
28         else
29             printf("Key not found\n");
30         char temp;
31         printf("Do you want to continue (y/n)? ");
32         getchar();
33         scanf("%c",&temp);
34         printf("%c",temp);
35         if(temp=='y'){
36             choice = 1;
37         }
38         else{
39             choice = 0;
40         }
41     }
42     return 0;
43 }
44
```

Output Snippet:

```
Enter the key to find : 4
Key found
Do you want to continue (y/n)? y
yEnter the key to find : 10
Key found
Do you want to continue (y/n)? y
yEnter the key to find : 43
Key not found
Do you want to continue (y/n)? n
n
...Program finished with exit code 0
Press ENTER to exit console.
```