```
1 /**
2 * OBLIG 2
3 *
 4 * THE PROGRAM:
 5 * -Set up variables and constants
 6 * -Create methods and functions
7 * -Optimize code
8 * -Use comments and documentation
9 *
10 * @file main.c
11 * @author Daniel AG
12 *
13 * Includes:
14 * - stdio.h for displaying and feeding data
15 * - stdbool.h for using booleans
16 * - ctype.h for toupper
17 */
18
19 ///Internal Includes
20 #include <stdio.h> //printf,scanf
21 #include <stdbool.h> //bool
22 #include <ctype.h> //toupper
23 #include <string.h> //strcpy,strcspn
24
25
26 /**
27 * @def MAXLANES
28 * @brief Maximum number of lanes
30 #define MAXLANES 18
31
32 /**
33 * @def MAXLANELENGTH
34 * @brief Maximum length of a lane
35 */
36 #define MAXLANELENGTH 100
37
38 /**
39 * @def MAXPARS
40 * @brief Maximum number of Pars
41 */
42 #define MAXPARS 8
43
44 /**
45 * @def STRLEN
46 * @brief Maximum length of a string (array of char)
47 */
48 #define MAXSTRLEN 100
49 /** @}*/
```

```
50
51
52 char laneDescription[MAXLANES][MAXSTRLEN] = { 0,0 }; // lane description
53 int laneLength[MAXLANES] = { 0 }; // lane length
54 int lanePar[MAXLANES] = { 0 }; // lane Par
55 bool laneOB[MAXLANES] = { 0 }; // lane OB
56 int numLanes = 0; // number of lanes
57
58
59 /**
60 * @def Run
61 * @brief Runs the program
62 * @return nothing
63 */
64 void Run();
65
66 /**
67 * @def Add
68 * @brief Adds a new lane
69 * @return true or false
70 */
71 void Add_Lane();
72
73 /**
74 * @def Display
75 * @brief Displays all lanes
76 * @return nothing
77 */
78 void Display_Lane();
79
80 /**
81 * @brief Executes the program
82 * @return int
83 */
84 int main()
85 {
       // Default values
86
87
       laneLength[0] = 62;
88
       lanePar[0] = 3;
89
       laneOB[0] = true;
       strcpy_s(laneDescription[0], MAXSTRLEN, "Lane with a lot of trees and >
90
         scrub");
91
92
       laneLength[1] = 94;
93
       lanePar[1] = 3;
94
       laneOB[1] = false;
95
       strcpy_s(laneDescription[1], MAXSTRLEN, "Flat terrain thourgout the
         map");
96
       numLanes = 2;
```

```
97
 98
        Run();
 99 }
100
101 void Run()
102 {
103
        char choice; // User choice for input
104
        do // conditional logic in while loop
105
        {
106
             /**
             * 1. Printing the menu
107
108
             * 2. Getting user input
109
             * 3. Activate switch case
110
             * 4. Returning to menu unless Q is pressed
111
             */
112
             printf("Menu Choices:\n");
             printf(" A - Add lane:\n");
113
             printf(" D - Display alle lanes:\n");
114
115
             printf(" Q - Quit:\n");
             printf(" Select a choice:");
116
117
             scanf_s(" %c", &choice, 1);
118
             choice = toupper(choice);
             printf("\n");
119
120
121
             switch (choice) // steps into corresponding case
122
             case 'A':
123
124
                 Add_Lane();
125
126
                 break;
127
             }
             case 'D':
128
129
130
                 Display_Lane();
131
                 break;
             }
132
             case 'Q':
133
134
135
                 printf("Quit - selected:\nEXITING PROGRAM");
136
                 break;
137
             default:
138
139
                 printf("Illegal argument");
140
141
         } while (choice != 'Q');
142 }
143
144 void Add_Lane()
145 {
```

```
146
         /**
147
             * 1. Takes inputs and generates new lanes
148
             * 2. The new lanes are added to the respective arrays
149
             * 3. After a new lane is created, increment number of lanes
150
         */
151
         if (MAXLANES <= numLanes) // is the maximum number of lanes reached?</pre>
152
153
             printf("[LOG]:Max Number of lanes created\n");
154
             return;
155
         }
156
157
         //Utility
158
         int currentLane = numLanes;
159
160
         // Lane data
161
         int qLaneLength = 0;
162
         int qLanePar = 0;
163
         char qLaneOB;
164
         char qLaneDescription[MAXSTRLEN];
165
         printf("How long is lane %i:", currentLane + 1);
166
         scanf_s("%d", &qLaneLength);
167
         if (qLaneLength <= 0) // flag if length is less or equal to 0</pre>
168
169
         {
             printf("Illegal argument..\nReturning\n\n");
170
171
             while (getchar() != '\n');
172
             return;
173
         // update lane length for current lane
174
175
         laneLength[currentLane] = qLaneLength;
176
177
         // Take input for lane Pars
178
         scanf_s("%d", &qLanePar);
179
         printf("Pars on the field. Choose a number bwteen (2-8):");
180
         if (MAXPARS < qLanePar) // flag if max Pars is overreached</pre>
181
             printf("Max Pars, set to MAXPARS\n");
182
183
             qLanePar = MAXPARS;
184
         if (qLanePar < 2) //flag if illegal argument</pre>
185
186
             printf("Illegal argument..\nReturning\n\n");
187
188
             while (getchar() != '\n');
189
             return;
190
         }
191
         // update lane Pars for current lane
192
         lanePar[currentLane] = gLanePar;
193
194
        // Take input for lane OB
```

```
...ering_H2025\PROG1001_H25\PROG1001_Oblig\Oblig2\main.c
```

```
195
        printf("Does the lane have OB (y = yes or n = no):");
196
        while (getchar() != '\n');
197
        scanf_s(" %c", &qLaneOB, 1);
        qLaneOB = toupper(qLaneOB);
198
        if (qLaneOB != 'Y' && qLaneOB != 'N') // flag if illegal argument
199
200
             printf("Illegal argument..\nReturning\n\n");
201
202
            while (getchar() != '\n');
203
            return;
204
        // update lane OB for current lane
205
        laneOB[currentLane] = (qLaneOB == 'Y');
206
207
        printf("Write a description:"); // Take input for lane description
208
209
        while (getchar() != '\n');
210
        fgets(qLaneDescription, MAXSTRLEN, stdin); // Read string with spaces
211
212
        // Remove newline character from string "\n"
        qLaneDescription[strcspn(qLaneDescription, "\n")] = 0;
213
214
        // copy the content of qLaneDescription to laneDescription
215
216
        strcpy_s(laneDescription[numLanes], MAXSTRLEN, qLaneDescription);
217
        numLanes++; //increment number of lanes
218
        printf("Lane %i added\n\n", currentLane); // confirm lane added
219
220 }
221
222 void Display_Lane()
223 {
224
        /**
            * 1. Iterate over all arrays and retieve content
225
226
            * 3. It then displays content
227
        */
228
229
        int totNumPars = 0; // total number of Pars
230
231
232
        for (int i = 0; i <= numLanes; i++) // iterate through all lanes</pre>
233
            if (numLanes == 0) // flag if no lanes are available
234
235
                 printf("No lanes available\n\n");
236
237
                continue; //skip empty lanes
238
            else if (laneLength[i] == 0) //flag if no data on lane
239
240
241
                 printf("No data on lane %d\n\n", i);
242
                continue;
243
            }
```

```
...ering_H2025\PROG1001_H25\PROG1001_Oblig\Oblig2\main.c
                                                                                 6
            printf("Lane \033[1;4m%d\033[0m:\n", i + 1);
244
            printf("
                         Length: \033[1;4m%d meters\033[0m\n", laneLength[i]);
245
246
            printf("
                          Pars: \033[1;4m%d\033[0m\n", lanePar[i]);
                         OB: %s\n", laneOB[i] ?
247
            printf("
                    "\033[1;4m With \033[0m" : "\033[1;4m Without \033[0m");
248
                          Description: \033[1;4m%s\033[0m\n\n",
249
250
                          laneDescription[i]);
251
            totNumPars += lanePar[i]; // sum up total number of Pars
252
        }
253
254
255
        printf("Summary\n");
256
        printf("Total number of lanes: \033[1;4m%d\033[0m\n", numLanes);
        printf("To get to Par, it requires number of throws: \033[1;4m%d\033
257
          [0m\n\n", totNumPars);
258 }
```

259