

```
1  /*@author Michał Pawłowski*/
2  /*Program należy wywoływać z 2 parametrami -i nazwaPlikuWejściowego -c
   standardowyZnakŁącucha */
3
4  #define _CRT_SECURE_NO_WARNINGS
5  #define _CRTDBG_MAP_ALLOC
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <stdbool.h>
10 #include <string.h>
11
12 /** Funkcja sprawdza parametry wejściowe oraz wczytuje z nich dane do
   zmiennych
13 @author Michał Pawłowski
14 @date 2020-07-05
15 @param argc ilość wczytanych parametrow
16 @param argv tablica wczytanych parametrow
17 @param input plik wejściowy
18 @param normalChar znak najczęściej występujący w ciągu
19 @return prawde jeżeli wczytano prawidłowo parametry
20 */
21 bool paramsCheck(int argc, char* argv[], char** input, char* normalChar) {
22
23     char* in = "-i";
24     char* nChar = "-c";
25     int count = 0;
26
27     if (argc == 5) {
28
29         for (int i = 1; i < argc; i++)
30         {
31             if (!strcmp(argv[i], in)) {
32
33                 *input = argv[i + 1];
34                 count++;
35             }
36
37             if (!strcmp(argv[i], nChar)) {
38
39                 *normalChar = *argv[i + 1];
40                 count++;
41             }
42         }
43
44         if (count == 2) return true;
45
46         else return false;
47     }
48
49     else return false;
50 }
51
```

```
52 /** Funkcja wypiuje pomoc
53 @author Michal Pawlowski
54 @date 2020-07-05
55 */
56 void help() {
57
58     printf("----- H E L P ----- \n");
59     printf("Program must be started with 2 parameters:\n");
60     printf("-i input file\n");
61     printf("-c standard char in array\n");
62     printf("----- \n");
63 }
64
65
66
67 int main(int argc, char* argv[]) {
68     {
69
70         char* input = NULL;
71         char normalChar;
72
73         if (!paramsCheck(argc, argv, &input, &normalChar)) {
74
75             help();
76         }
77
78         else {
79
80             FILE* file = fopen(input, "r");
81             if (file) {
82
83                 char sign;
84                 int count = 0, last = -1;
85                 bool isTitleDisplayed = false;
86
87                 while ((sign = getc(file)) != EOF) {
88
89                     if (sign == normalChar) count++;
90                     else {
91
92                         if (!isTitleDisplayed) {
93
94                             fprintf(stdout, "Found spacial chars in array of ↗
95                             %c:\n\n", normalChar);
96                             isTitleDisplayed = true;
97                         }
98
99                         if (last < 0) {
100                             fprintf(stdout, "- \"%c\" -> on position: %d\n", ↗
101                             sign, count);
102                             last = count;
103                             count++;
104                         }
105                     }
106                 }
107             }
108         }
109     }
110 }
```

```
103         }
104         else {
105
106             fprintf(stdout, "-\"%c\" -> on position: %d ->
from last special char: +%d\n", sign, count, count -
last);
107             last = count;
108             count++;
109         }
110     }
111 }
112
113 fclose(file);
114
115 if (!isTitleDisplayed) fprintf(stdout, "Special chars not
found in array!");
116
117 }
118
119
120 _CrtDumpMemoryLeaks();
121 return 0;
122 }
123 }
```