

Jan 07, 14 14:47

theBomb.c

Page 1/4

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 void explode(const int code);
6
7 void readInput(char buffer[], const int bufferSize){
8     int i;
9     int c;
10    for(i=0; i<=bufferSize; ++i){
11        c = getchar();
12
13        /* this checks that you are at the end of the line */
14        /* Windows encodes an end-of-line as two characters: \r\n */
15        /* Linux just uses \n */
16        /* This will accept either version */
17        if(c == '\r'){
18            c = getchar();
19        }
20
21        if(c == '\n'){
22            break;
23        }
24        else if(i<bufferSize){
25            buffer[i] = (char)c;
26        }
27    }
28 }
29
30 void part1(){
31     const int bufferSize = 12;
32     char buffer[bufferSize];
33     char target[] = "Careful";
34     int i;
35     const int n = strlen(target);
36
37     readInput(buffer, bufferSize);
38
39     for(i=0; i<n; i++){
40         if(buffer[i] != target[i]){
41             explode(1);
42         }
43     }
44 }
45
46 int part2(){
47     int i,z;
48
49     const int bufferSize = 12;
50     char buffer[bufferSize];
51
52     readInput(buffer, bufferSize);
53
54     /* atoi takes a string representation of a number (e.g. "1234") */
55     /* and converts it to an int representation of the number (e.g. 1234) */
56     z = atoi(buffer);
57     if(!z)
58         explode(20);
59
60     z += 17;
61
62     if(z != 93)
63         explode(21);
64
65     return z / 2;
66 }
67
68 void part3(){
69     int i,z;
70
71     const int bufferSize = 12;
72     char buffer[bufferSize];
73 
```

Jan 07, 14 14:47

theBomb.c

Page 2/4

```

74     readInput(buffer, bufferSize);
75
76     i=0;
77     while(i < bufferSize && (buffer[i] == 'Z' || buffer[i] == 'j')){
78         i++;
79     }
80
81     if(i != 3)
82         explode(70);
83 }
84
85 void part4(const int x){
86     int i,y,z;
87     const int bufferSize = 12;
88     char buffer[bufferSize];
89
90     readInput(buffer, bufferSize);
91
92     /* atoi takes a string representation of a number (e.g. "1234") */
93     /* and converts it to an int representation of the number (e.g. 1234) */
94     z = atoi(buffer);
95     if(!z)
96         explode(20);
97
98     y = z;
99     for(i=0; i<50; i++){
100         y += z * i * y;
101     }
102
103     if(z + 9 != x){
104         explode(21);
105     }
106 }
107
108 void part5(char buffer[], const int bufferSize){
109     const char allowed[] = "az3hoPeACce6";
110     int i, j, x;
111
112     for(i=0; i<bufferSize; ++i){
113         char c;
114
115         x = getchar();
116         if(x == EOF)
117             explode(10);
118         else if(x == '\n')
119             explode(11);
120
121         c = (char)x;
122         for(j=2; j<10; ++j){
123             if(c == allowed[j]){
124                 buffer[i] = c + 1;
125                 break;
126             }
127         }
128         if(j == 10){
129             explode(12);
130         }
131     }
132
133     j = 0;
134     for(i=1; i<bufferSize; ++i){
135         if(buffer[i] != buffer[i-1]){
136             j = 1;
137             break;
138         }
139     }
140     if(j == 0)
141         explode(13);
142
143     /* this checks that you are at the end of the line */
144 
```

Jan 07, 14 14:47

theBomb.c

Page 3/4

```

147  /* Windows encodes an end-of-line as two characters: \r\n */
148  /* Linux just uses \n */
149  /* This will accept either version */
150  x = getchar();
151  if(x == '\r')
152      x = getchar();
153  if(x != '\n')
154      explode(14);
155  }
156
157
158  void part6(const char buf1[], const int bufSize){
159      char buf2[bufSize];
160      int i, c;
161
162      readInput(buf2, bufSize);
163
164      for(i=0; i<bufSize; ++i){
165          if(buf1[i] != buf2[bufSize - 1 - i])
166              explode(-44);
167      }
168  }
169
170
171  void part7(int *x){
172      int y = 29;
173      int *z = &y;
174      int k;
175
176      const int bufSize = 12;
177      char buffer[bufSize];
178
179      *z -= 3;
180
181      readInput(buffer, bufSize);
182
183      /* atoi takes a string representation of a number (e.g. "1234") */
184      /* and converts it to an int representation of the number (e.g. 1234) */
185      k = atoi(buffer);
186
187      if(k + *x != y)
188          explode(70);
189  }
190
191
192  void theBomb(){
193      const int bufSize = 10;
194      char buffer[bufSize];
195      int result;
196
197      printf("Enter the stage 1 password: ");
198      part1();
199      printf("Stage 1 defused.\n");
200
201      printf("Enter the stage 2 password: ");
202      result = part2();
203      printf("Stage 2 defused.\n");
204
205      printf("Enter the stage 3 password: ");
206      part3();
207      printf("Stage 3 defused.\n");
208
209      printf("Enter the stage 4 password: ");
210      part4(result);
211      printf("Stage 4 defused.\n");
212
213      printf("Enter the stage 5 password: ");
214      part5(buffer, bufSize);
215      printf("Stage 5 defused.\n");
216
217      printf("Enter the stage 6 password: ");
218      part6(buffer, bufSize);
219      printf("Stage 6 defused.\n");

```

Jan 07, 14 14:47

theBomb.c

Page 4/4

```

220
221      printf("Enter the stage 7 password: ");
222      part7(&result);
223      printf("Stage 7 defused.\n");
224
225      printf("The entire bomb has been defused.\n");
226  }
227

```