

Estrutura de Dados 1 – Correção da questão 13 – P1

Nome: Pedro Igor Oliveira Silva **Matrícula:** 17/0062635

O código enviado como solução da questão:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 long int words(const char *in) {
5
6     if (in == NULL) {
7         return -1;
8     }
9
10    FILE *arq = fopen(in, "r");
11
12    if (arq == NULL) {
13        return -2;
14    }
15
16    long int tamanho;
17
18    while (fgetc(arq) != EOF);
19
20    tamanho = ftell(arq);
21
22    fclose(arq);
23
24    return tamanho;
25 }
```

Meu erro foi de leitura e interpretação. No lugar de implementar um algoritmo que conte a quantidade de palavras de um arquivo, implementei um algoritmo que conte a quantidade de caracteres de um arquivo.

O erro se encontra a partir da linha 18 até a linha 20.

Para corrigir criei duas variáveis do tipo char. Uma para armazenar o valor atual do fgetc(arq) que retorna caracteres na ordem do arquivo. E outra para armazenar o caractere da leitura anterior. Dessa forma, se o caractere atual for um espaço ou fim de arquivo e o caractere anterior um caractere diferente de espaço temos aí uma palavra.

Confira na próxima página a resolução pós-correção.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int words(const char *in){
4      if(in == NULL){
5          return -1;
6      }
7
8      FILE *arq = fopen(in, "r");
9
10     if(arq == NULL){
11         return -2;
12     }
13
14     char aux = ' ', prev;
15     int qtd = 0;
16     while(1){
17         prev = aux;
18         aux = fgetc(arq);
19         if((aux == ' ' || aux == EOF) && (prev != ' ')){
20             qtd++;
21         }
22         if(aux == EOF){
23             break;
24         }
25     }
26     fclose(arq);
27     return qtd;
28 }

```

```

#include <stdio.h>
#include <stdlib.h>
int words(const char *in){
    if(in == NULL){
        return -1;
    }

    FILE *arq = fopen(in, "r");

    if(arq == NULL){
        return -2;
    }
    char aux = ' ', prev;
    int qtd = 0;
    while(1){
        prev = aux;
        aux = fgetc(arq);
        if((aux == ' ' || aux == EOF) && (prev != ' ')){
            qtd++;
        }
        if(aux == EOF){
            break;
        }
    }
    fclose(arq);
    return qtd;
}

```