

## Sprawdzanie 1

Szymon Barczyk

Zadania do wykonania:

1. Program musi być uruchamiany z dwoma argumentami z linii komend. W przeciwnym wypadku powinien poinformować o błędzie i zakończyć się.
2. Sprawdzić, czy istnieje plik o nazwie „tmp.txt” i czy jest on możliwy do odczytania. Jeżeli plik nie istnieje lub proces nie ma dostępu do odczytu, wówczas program powinien o tym poinformować i zakończyć się.
3. Wprowadzony pierwszy argument należy potraktować jako nazwę pliku. Należy sprawdzić czy plik o takiej nazwie istnieje:
  - jeśli istnieje:  
należy sprawdzić prawo zapisu do pliku. Jeśli proces nie ma prawa zapisu do tego pliku wówczas poinformować o tym i zakończyć program. Jeśli proces ma prawo zapisu do tego pliku wówczas należy go nadpisać.
  - jeśli nie istnieje:  
należy stworzyć plik z prawami zapisu i odczytu przez wszystkich użytkowników (0666)

```
9      FILE *oldFile;
10     FILE *newFile;
11
12     // 1. Sprawdź ilość argumentów
13     if (argc < 3)
14     {
15         perror("Nie podano dwóch argumentów!");
16         exit(1);
17     }
18
19     // 2. Sprawdź dostęp do "tmp.txt"
20     oldFile = fopen("tmp.txt", "r");
21     if (oldFile == NULL)
22     {
23         perror("Nie udało się odczytać pliku tmp.txt!");
24         exit(2);
25     }
26
27     // 3. Sprawdź dostęp do nowego pliku
28     newFile = fopen(argv[1], "w+");
29     if (newFile == NULL)
30     {
31         perror("Nie udało się otworzyć nowego pliku!");
32         exit(3);
33     }
```

4. Zapisać do pliku w pierwszej linii: „zadanie 1:”  
Odczytać 3, 5, 9 linię tekstu z pliku „tmp.txt” i wypisać je  
na ekranie oraz przepisać je do nowego pliku.

```
35 // 4.  
36 char *tmp = NULL;  
37 size_t len = 120;  
38 fprintf(newFile, "Zadanie 1:\n");  
39 getline(&tmp, &len, oldFile);  
40 getline(&tmp, &len, oldFile);  
41 getline(&tmp, &len, oldFile);  
42 fprintf(newFile, tmp);  
43 getline(&tmp, &len, oldFile);  
44 getline(&tmp, &len, oldFile);  
45 fprintf(newFile, tmp);  
46 getline(&tmp, &len, oldFile);  
47 getline(&tmp, &len, oldFile);  
48 fprintf(newFile, tmp);  
49 fseek(oldFile, 0, SEEK_SET);
```

5. Zapisać do pliku w nowej linii: „zadanie 2:”  
Odczytać 2, 4, 7 linię tekstu z pliku „tmp.txt”, zamienić wszystkie małe litery na duże litery oraz przepisać je do nowego pliku.

```
51 // 5.
52 fprintf(newFile, "Zadanie 2:\n");
53 getline(&tmp, &len, oldFile);
54 getline(&tmp, &len, oldFile);
55 for(int i = 0; i < strlen(tmp); i++){
56     tmp[i] = toupper(tmp[i]);
57 }
58 fprintf(newFile, tmp);
59 getline(&tmp, &len, oldFile);
60 getline(&tmp, &len, oldFile);
61 for(int i = 0; i < strlen(tmp); i++){
62     tmp[i] = toupper(tmp[i]);
63 }
64 fprintf(newFile, tmp);
65 getline(&tmp, &len, oldFile);
66 getline(&tmp, &len, oldFile);
67 for(int i = 0; i < strlen(tmp); i++){
68     tmp[i] = toupper(tmp[i]);
69 }
70 fprintf(newFile, tmp);
71 fseek(oldFile, 0, SEEK_SET);
```

6. Zapisać do pliku w nowej linii: „zadanie 3:”  
Przepisać wszystkie cyfry występujące w pliku „tmp.txt” do nowego pliku.

```
73 // 6.
74 fprintf(newFile, "Zadanie 3:\n");
75 int k;
76 do {
77     char numbers[100] = "";
78     k = getline(&tmp, &len, oldFile);
79     int j = 0;
80     for(int i = 0; i < strlen(tmp); i++){
81         if(isdigit(tmp[i])){
82             numbers[j] = tmp[i];
83             j++;
84         }
85     }
86     fprintf(newFile, "%s\n", j, numbers);
87 }while(k >= 0);
88 fseek(oldFile, 0, SEEK_SET);
```

7. Zapisać do pliku w nowej linii: „zadanie 4:”  
Porównać tekst z pliku „tmp.txt” z drugim argumentem i zapisać do nowego pliku położenie identycznych ciągów znaków.

```
90 // 7.
91 fprintf(newFile, "Zadanie 4:\n");
92 int line = 0;
93 do {
94     line++;
95     k = getline(&tmp, &len, oldFile);
96     if(strstr(tmp, argv[2]) != NULL){
97         fprintf(newFile, "Linia nr: %i ", line);
98     }
99 } while (k >= 0);
100 ...
```

8. Wypisać na ekranie wielkość pliku.
9. Zmienić prawa do pliku na (0600).
10. Zamknąć otwarte pliki.

```
102 // 9.  
103 chmod(argv[1], 0600);  
104  
105 // 10. Zamknij pliki  
106 fclose(oldFile);  
107 fclose(newFile);  
108  
109 // 8.  
110 struct stat sb;  
111 if(stat(argv[1], &sb) != -1){  
112     printf("Wielkość pliku \"%s\" wynosi %d", argv[1], sb.st_size);  
113 }
```

tutaj zrzut ekranu utworzonego pliku txt  
lub wklejona jego zawartość

```
1 Zadanie 1:  
2 Mój numer telefonu to 693-487-225.  
3 Kurs waluty wynosił dziś 4.53 PLN za 1 euro.  
4 Lot o numerze 758 wystartował o 16:45.  
5 Zadanie 2:  
6 SPOTKANIE ODBĘDZIE SIĘ 12 PAŹDZIERNIKA 2024 ROKU.  
7 W SALI BYŁO 25 OSÓB, W TYM 9 DZIECI.  
8 NUMER POKOJU W HOTELU TO 304.  
9 Zadanie: 3  
10 5129  
11 122024  
12 693487225  
13 259  
14 4531  
15 304  
16 7581645  
17 7  
18 8814  
19 18  
20 3  
21 452  
22 237  
23 12  
24 6  
25 6  
26 Zadanie 4:  
27 Linia nr: 3 Linia nr: 7
```