Iva Pleić - 3. zadatak

Reviewing last commited code version: 210e692
Reviewed by Kristina Zoë Mustapić

1) U funkciji **SortList**, iako to nije krucijalno za ispravno izvršavanje zadatka funkcije, prema prototipu funkcije, vraća tip podatka **int**, ne zaboravljati naredbu: return EXIT_SUCCESS; Zbog maksimalnog oslobađanja nepotribne memorije, free-at svaki element liste po završetku izvršavanja funkcije, gdje je to logički izvedivo. (vidljivo iz prototipa funkcije)
Npr. kod f-ja gdje kreiramo/inicijaliziramo podatke za određeni element liste vraćamo/"return-amo" pokazivač na taj element listu, tkd je ne možemo „free-at"
Npr. `PersonP CreatePerson(char* firstName, char* lastName, int birthYear);`

   Dok npr. Kod funkcija di su korišteni "pomoćni" elementi (temp – "temporary"), a funkcija NE vraća pokazivač na element liste, treba ih "free-at" po završetku njihova korištenja.
   Well explained: https://stackoverflow.com/questions/67084229/invalid-free-error-when-freeing-memory-of-a-singly-linked-list

   Npr. `int SortList(PersonP head); -> ISPRAVNO: objašnjeno u komentarima koda + link!`

```
int SortList(PersonP p)
{
    PersonP a = NULL, b = NULL, prev = NULL, end = NULL;
    PersonP temp;
    for (a = p; a->next != end;)
    {
        prev = a;
        for (b = a->next; b->next != end; b = b->next)
        {
            if (strcmp(b->lastName, b->next->lastName) > 0)
            {
                temp = b->next;
                prev->next = temp;
                b->next = temp->next;
                temp->next = b;
                b = temp;
            }
            prev = b;
        }
        end = b;
    }
    free(a);
    free(b);
    free(prev);
    free(end);
    free(temp);

    return EXIT_SUCCESS;
}
```

2) U case-u **9**, ponavljanje greške iz prošlog zadatka 😐 plaky plaky
   Kad imaš više uzastopnih printf-ova, nastojat ih spajat u jedinstveni poziv funckcije scanf (string concatenation). Želiš maksimalno ubrzat izvršavanje programa. Svaki dodatni poziv produžava izvršavanje programa, detaljnije pročitat u review-u od 2. zadatka.

```
        break;

    case 9:
        printf(
            "========================================================================\n"
            "You chose to insert the list of people in the file:\n\n");
        fileName = EnterFileName();
        WriteInFile(p, fileName);
        printf(
            "\nList has been sucesfully added to the file : %s\n\n", fileName);
        printf("\nPress enter to continue app execution.\n"
            "========================================================================\n");
        system("pause");
        break;
```

```
    case 9:
        printf(
            "========================================================================\n"
            "You chose to insert the list of people in the file:\n\n");
        fileName = EnterFileName();
        WriteInFile(p, fileName);
        printf(
            "\nList has been sucesfully added to the file : %s\n\n"
            "\nPress enter to continue app execution.\n"
            "========================================================================\n", fileName);
        system("pause");
        break;
```

3) U funkciji **EnterFileName** poželjno je ukomponirat funkciju **EnterValidString,** da bi se onemogućia unos praznog stringa.

4) Pravilno napisat su **CC** e **SS** fu **LL** y ☺

5) Stvorit naviku da pokazivač na file imenovat smisleno fp („file pointer“)

6) Konzistentno „return-at“ ili EXIT_FAILURE ili EXIT_SUCCESS, izbjegavat: return -1; (u funkciji **WriteInFile**)