

Технически университет – София

ПИК1

Курсова работа

Разработил:
Денис Сабри
ф.н. 121219118

Одобрил:
гл. ас. д-р Елица Гиева
13.01.2020

Функции в програмата:

calloc - Функцията `calloc ()` заделя памет големината на която е равна `num * size` . Така `calloc ()` заделя достатъчно памет за масив от `num` обекта с размер `size` . Функцията `calloc ()` връща указател към първия байт на заделената регион

malloc - Функцията `malloc ()` връща указател към първия байт от регион в паметта с размер `size` който е бил заделен от хийпа . Ако има недостатъчно памет в хийпа за изпълнение на заявката `malloc ()` връща нулев указател .

strlen - Функцията `strlen` изчислява дължина на стринг. Стойността на върнатия резултат от функция `strlen` е целочислен тип `int`.

strncat - Тази функция има два аргумента: низ, към който ще се добавя и низ, който ще се добавя.

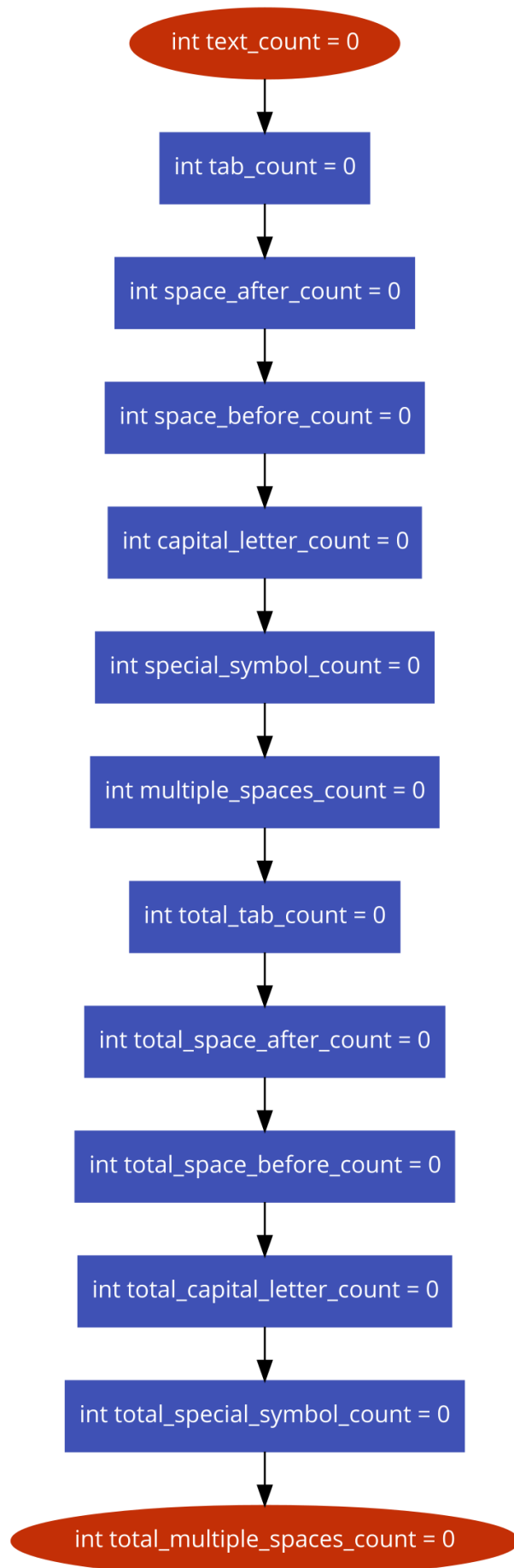
sizeof - По време на компилирането библиотечната функция `sizeof` връща дължината на заеманата оперативна памет от дадена променлива. Всеки един от вградените типове променливи – `char`, `int`, `float`, `double` заема точно определен брой бита памет.

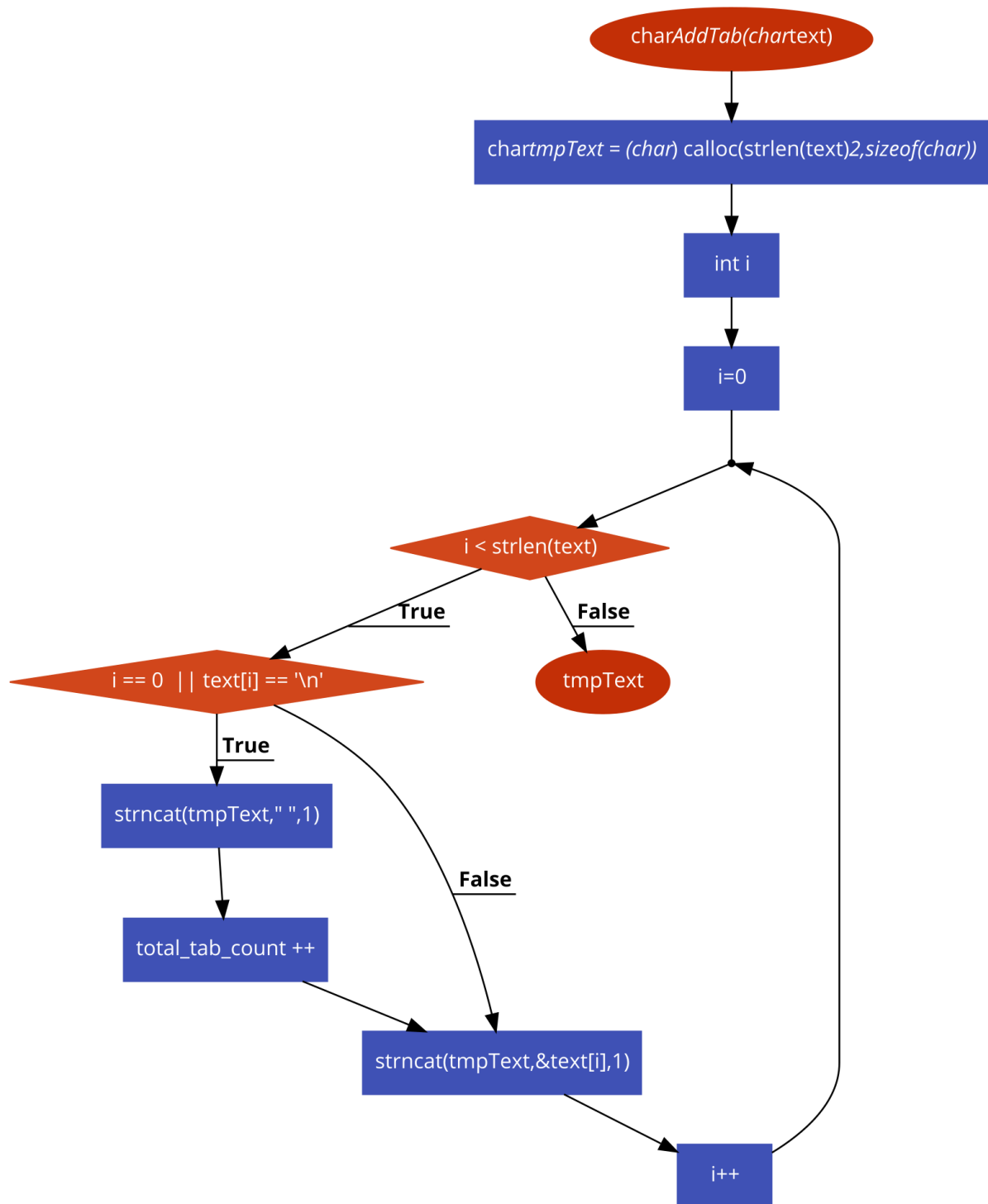
toupper – Тази функция преобразува една, отделна латинска буква главна в малка и обратно. Аргументът е от тип `char`.

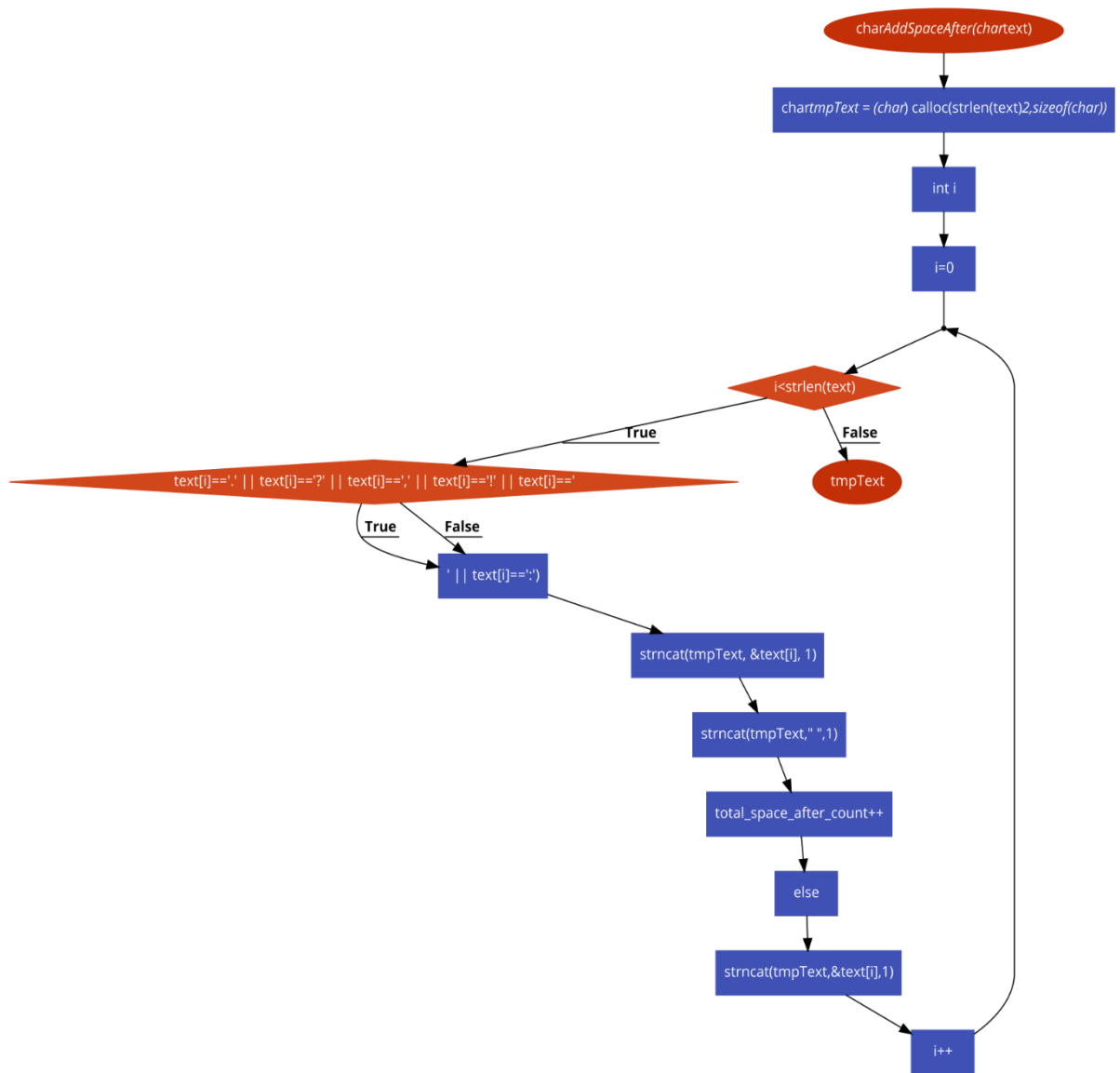
fflush(stdin) – Това е така наречената функция за защитено четене на входни данни т.е. когато имам `"scanf("%d",&a)"`, например, да не дава на потребителя да въвежда нищо друго, освен цифри (ако въведе символ, да го върне обратно).

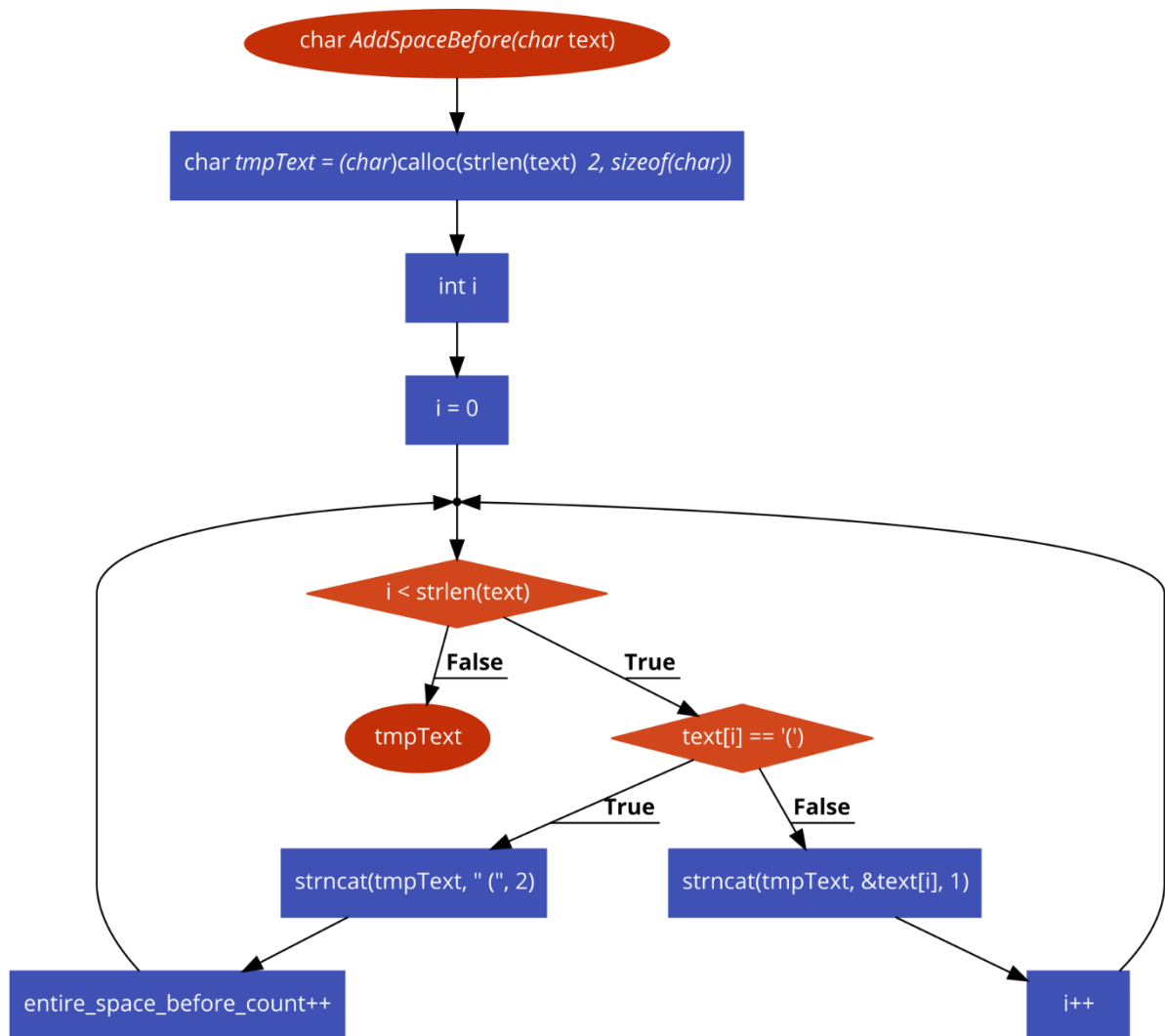
ограничава максималната възможна дължина в произволен ред т.е. най-големия брой символи в един ред от този файл.

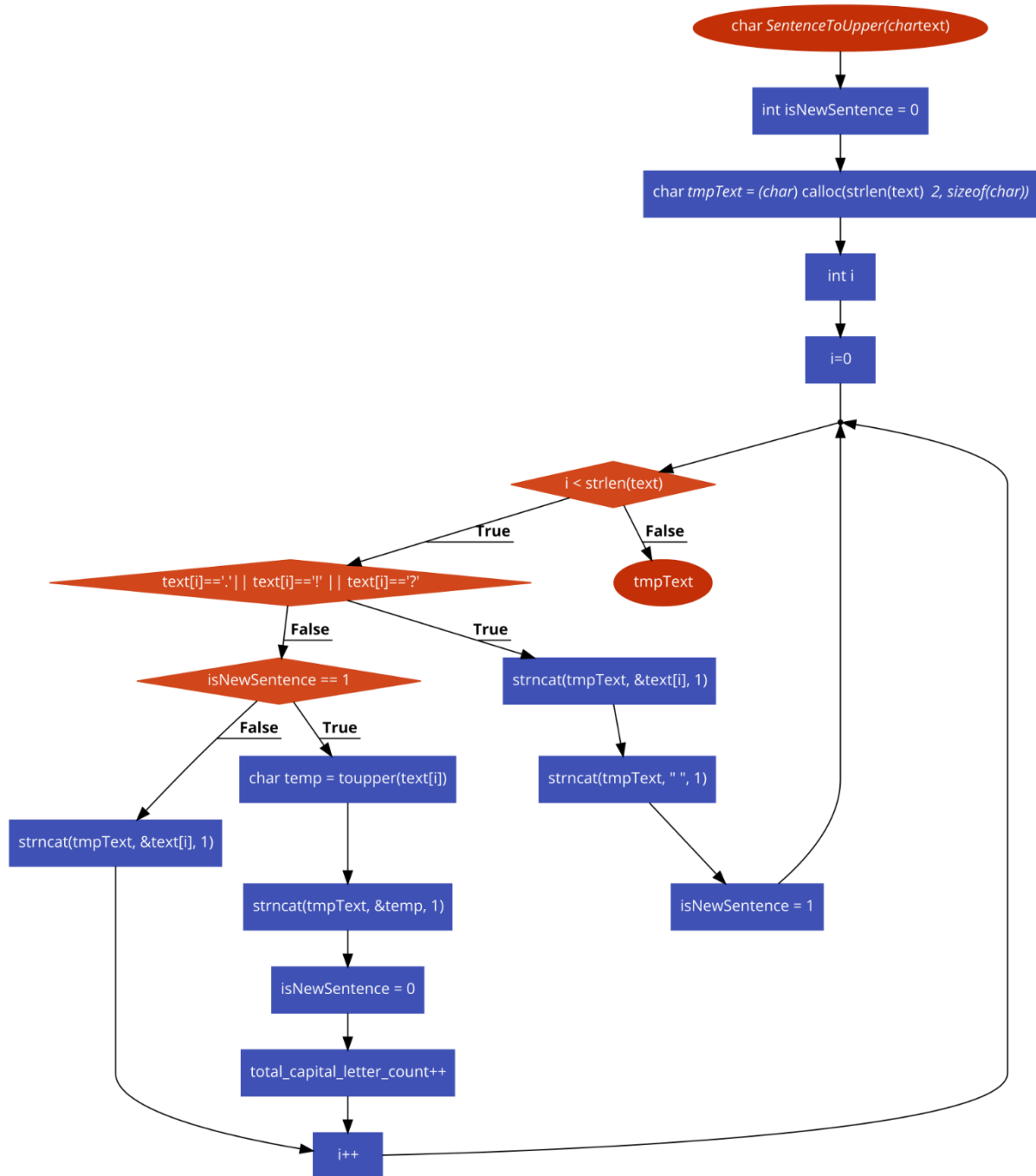
strcpy – осъществява се копиране на знаци от стринг.

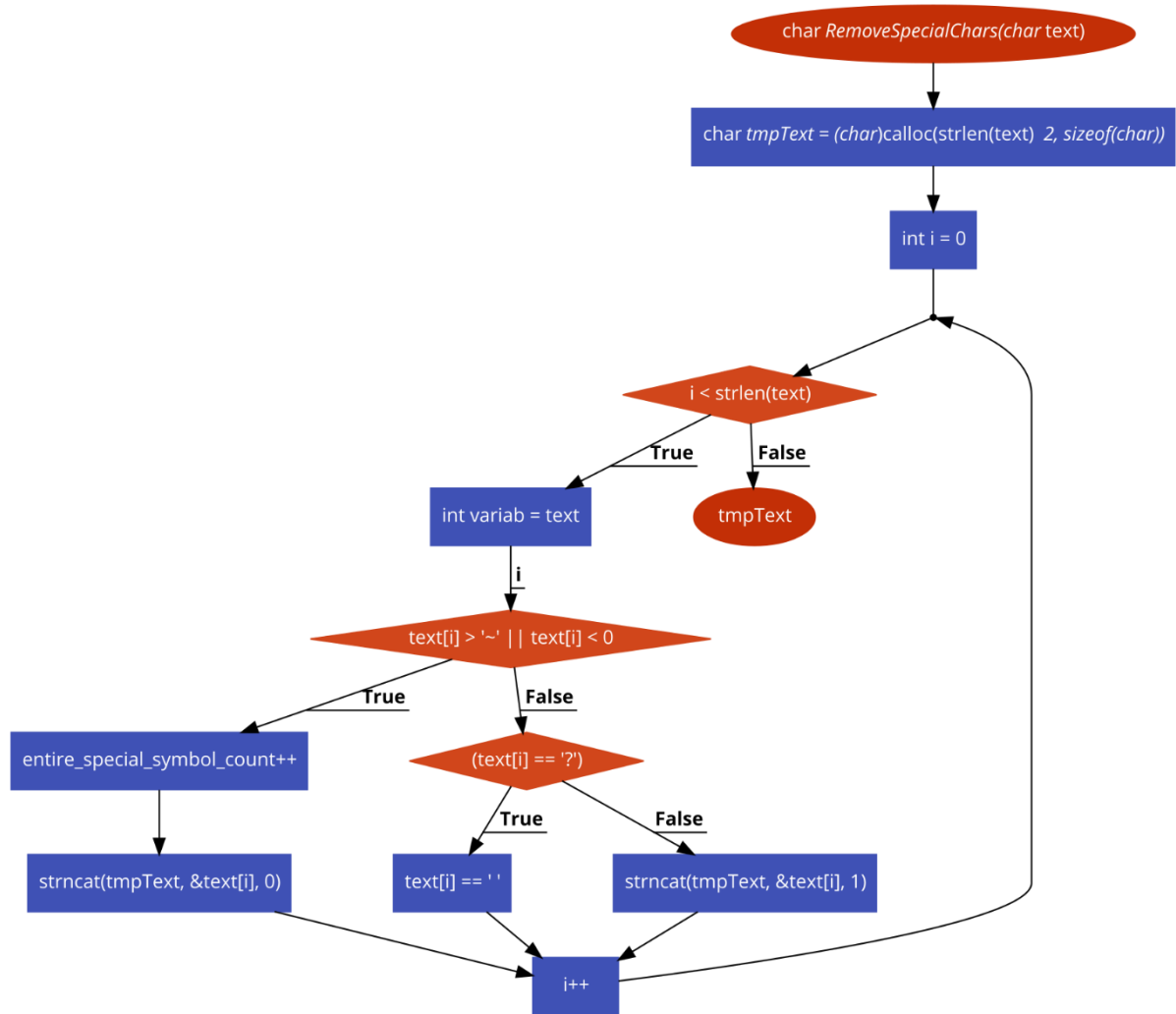


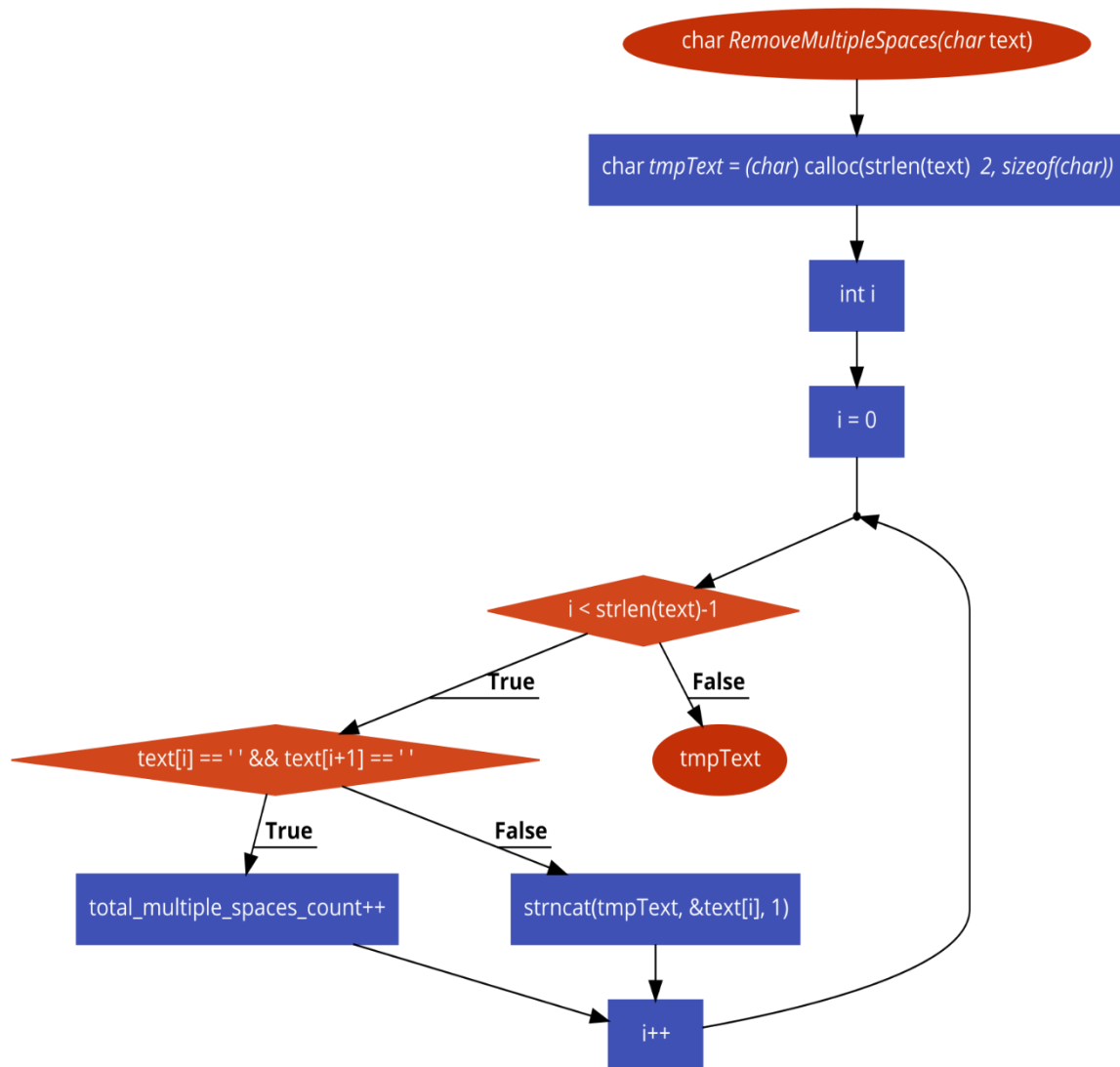


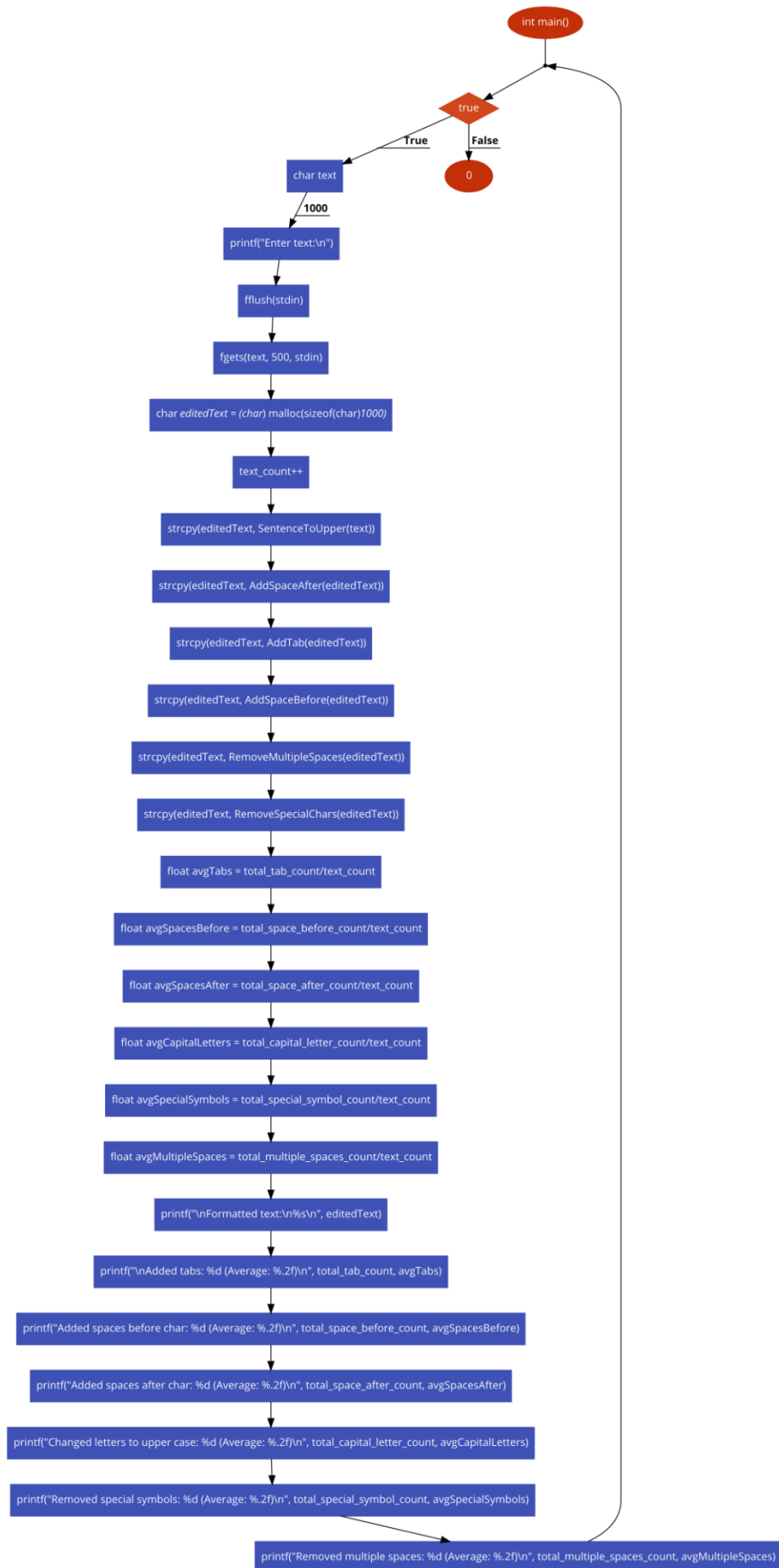












```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdbool.h>
#include <string.h>

int text_count = 0;

int tab_count = 0;
int space_after_count = 0;
int space_before_count = 0;
int capital_letter_count = 0;
int special_symbol_count = 0;
int multiple_spaces_count = 0;

int entire_tab_count = 0;
int entire_space_after_count = 0;
int entire_space_before_count = 0;
int entire_capital_letter_count = 0;
int entire_special_symbol_count = 0;
int entire_multiple_spaces_count = 0;

char* AddTab(char* text) { // sled vseki abzac izrechenieto zapochva s 1
    tabulaciq navutre

    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char)); /*
    int i;

    for (i = 0; i < strlen(text); i++) {

        if (i == 0 || text[i] == '\n') {
            strcat(tmpText, "\t", 1);
            entire_tab_count++;
        }
        strcat(tmpText, &text[i], 1);
    }
    return tmpText;
}

char* AddSpaceAfter(char* text) { //sled vseki prepinatelen znak funkciqta
    dobavq po 1 shpaciq

    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char));
    int i;

    for (i = 0; i < strlen(text); i++) {

        if (text[i] == '.' || text[i] == '?' || text[i] == ',' || text[i]
        == '!' || text[i] == ';' || text[i] == ':' || text[i] == '"') {

            strcat(tmpText, &text[i], 1);
            strcat(tmpText, " ", 1);

            entire_space_after_count++;

        }
        else {
            strcat(tmpText, &text[i], 1);
        }
    }
    return tmpText;
}

```

```

}

char* AddSpaceBefore(char* text) { // funkciqta dobavq shpaciq predi
lqvata skoba
    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char));
    int i;

    for (i = 0; i < strlen(text); i++) {
        if (text[i] == '(')
        {
            strncat(tmpText, " (", 2);
            entire_space_before_count++;
            continue;
        }
        strncat(tmpText, &text[i], 1);
    }
    return tmpText;
}

char* SentenceToUpper(char* text) { // vsqko izrechenie zapochva s glavna
bukva

    int isNewSentence = 0;

    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char));
    int i;

    for (i = 0; i < strlen(text); i++) {

        if (text[i] == '.' || text[i] == '!' || text[i] == '?') {
            strncat(tmpText, &text[i], 1);
            strncat(tmpText, " ", 1);
            isNewSentence = 1;
            continue;
        }

        if (isNewSentence == 1) {

            char temp = toupper(text[i]);

            strncat(tmpText, &temp, 1);
            isNewSentence = 0;

            entire_capital_letter_count++;
        }
        else {
            strncat(tmpText, &text[i], 1);
        }
    }
    return tmpText;
}

char* RemoveSpecialChars(char* text) { //vseki simvol sys stoynost po-
golqma ot 127 biva premahvan
    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char));

    for (int i = 0; i < strlen(text); i++) {
        int variab = text[i];

        if (text[i] > '~' || text[i] < 0) {

```

```

        entire_special_symbol_count++;
        strncat(tmpText, &text[i], 0);
    }
    else if ((text[i] == '?')){
        text[i] == ' ';
    }else
    {
        strncat(tmpText, &text[i], 1);
    }
}
return tmpText;
}

char* RemoveMultipleSpaces(char* text) { //prazni razstojniq po-golemi ot
1 shpaciq bivat namaleni do 1
    char* tmpText = (char*)calloc(strlen(text) * 2, sizeof(char));
    int i;

    for (i = 0; i < strlen(text) - 1; i++) {

        if (text[i] == ' ' && text[i + 1] == ' ')
        {
            entire_multiple_spaces_count++;
        }
        else {
            strncat(tmpText, &text[i], 1);
        }
    }
    return tmpText;
}

int main()
{
    while (true) {
        char text[2000];

        printf("Enter text:\n");
        fflush(stdin);
        fgets(text, 1000, stdin);

        char* editedText = (char*)malloc(sizeof(char) * 2000);
        text_count++;

        strcpy(editedText, RemoveSpecialChars(text));
        strcpy(editedText, SentenceToUpper(editedText));
        strcpy(editedText, AddSpaceAfter(editedText));
        strcpy(editedText, AddTab(editedText));
        strcpy(editedText, AddSpaceBefore(editedText));
        strcpy(editedText, RemoveMultipleSpaces(editedText));

        float averageTabs = entire_tab_count / text_count;
        float averageSpacesBefore = entire_space_before_count / text_count;
        float averageSpacesAfter = entire_space_after_count / text_count;
        float averageCapitalLetters = entire_capital_letter_count /
text_count;
        float averageSpecialSymbols = entire_special_symbol_count /
text_count;

```

```
        float averageMultipleSpaces = entire_multiple_spaces_count /
text_count;

        printf("\nFormatted text:\n%s\n", editedText);
        printf("\nAdded tabs: %d (Average: %.2f)\n", entire_tab_count,
averageTabs);
        printf("Added spaces before char: %d (Average: %.2f)\n",
entire_space_before_count, averageSpacesBefore);
        printf("Added spaces after char: %d (Average: %.2f)\n",
entire_space_after_count, averageSpacesAfter);
        printf("Changed letters to upper case: %d (Average: %.2f)\n",
entire_capital_letter_count, averageCapitalLetters);
        printf("Removed special symbols: %d (Average: %.2f)\n",
entire_special_symbol_count, averageSpecialSymbols);
        printf("Removed multiple spaces: %d (Average: %.2f)\n",
entire_multiple_spaces_count, averageMultipleSpaces);

    }
    return 0;
}
```


Курсова работа по дисциплина ПИК1 за студенти ФКСТ

Да се напише програма на C, която имитира работата на текстов редактор. При въведен текст от потребителя, програмата трябва да върне обратно съответния текст коригиран. Спазват се следните правила:

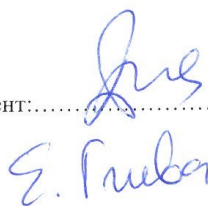
1. След всеки абзац изречението започва с 1 табулация навътре;
2. След всеки препинателен знак програмата поставя 1 шпация (при срещане на скоби поставя шпация преди лява и не поставя след дясна);
3. Всяко изречение трябва да започва с главна буква.
4. Всеки нестандартен символ бива премахван(символ със стойност по-голяма от 127);
5. Празни разстояния по-големи от 1 шпация да бъдат намалени до 1 шпация.

След като се принтира коригираният текст да се изведе броя на коригираните грешки и да се изведе в проценти броят на грешките спрямо средните грешки на всеки 1 текст въведен от началото на програмата.

Студент:.....



асистент:.....



E. Губа

Вход:

Star Wars is an American epic space-opera media franchise created by George Lucas, which began with the eponymous (1977) film and quickly became a worldwide pop-culture phenomenon. The franchise has been expanded into various films and other media, including television series, video games, novels, comic books, theme park attractions, and themed areas, comprising an all-encompassing fictional universe. The franchise holds a Guinness World Records title for the "Most successful film merchandising franchise", in 2018, the total value of the *Star Wars* franchise was estimated at US\$65 billion, and it is currently the fifth-highest-grossing media franchise of all time.

Изход:

Star Wars is an American epic space-opera media franchise created by George Lucas, which began with the eponymous (1977) film and quickly became a worldwide pop-culture phenomenon. The franchise has been expanded into various films and other media, including television series, video games, novels, comic books, theme park attractions, and themed areas, comprising an all-encompassing fictional universe. The franchise holds a Guinness World Records title for the "Most successful film merchandising franchise", in 2018, the total value of the Star Wars franchise was estimated at US\$65 billion, and it is currently the fifth-highest-grossing media franchise of all time.

LABS:

<ctype.h> - Библиотека за тестване и промяна на отделни символи в низ

<string.h> - Библиотека за низови функции, манипулиране на низове.

<stdbool.h> в стандартната библиотека на C за езика за програмиране на C съдържа четири макроса за булев тип данни.