#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TEXT\_SIZE 1

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

//возможно стоит вынести из функции

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

char end = str[i - 1];

return end;

}

int main() {

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

char\*\* text = malloc(1 \* sizeof(char\*)); // задание памяти под текст

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

if (check != '?') {

char\*\* text\_big = realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

//for ( int i = 0; i < (j + TEXT\_SIZE); i++) {

//free (text\_big[i]);

//}

text[j] = new\_str; // надо увеличить текст

count\_first++;

count\_end++;

j++;

}

else {

count\_first++;

}

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

for (int k = 0; k < j; k++) {

printf("%s\n", text[k]);

}

free(text);// освободить память !! хз пока как но разберешься

return 0;

}

Последний относительно рабочий код:  
#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TEXT\_SIZE 1

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

//возможно стоит вынести из функции

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

char end = str[i - 1];

return end;

}

int main() {

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

char\*\* text = malloc(1 \* sizeof(char\*)); // задание памяти под текст

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

if (check != '?') {

if (j > 0) {

char\*\* text\_big = realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

}

//for ( int i = 0; i < (j + TEXT\_SIZE); i++) {

//free (text\_big[i]);

//}

text[j] = new\_str; // надо увеличить текст

count\_first++;

count\_end++;

j++;

}

else {

count\_first++;

}

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

for (int k = 0; k < j; k++) {

printf("%s\n", text[k]);

}

printf("Количество предложений до %d и количество предложений после %d", count\_first - 1, count\_end);// вывести массив

for (int l = 0; l < count\_end; l++) {

free(text[l]);

}

free(text);// освободить память !! хз пока как но разберешься

return 0;

}

Код с проверками  
#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TEXT\_SIZE 1

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

char end = str[i - 1];

printf("[%c]", end);

return end;

}

int main() {

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

char\*\* text = malloc(TEXT\_SIZE \* sizeof(char\*));

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

printf("(%s)", new\_str);

if (check != '?') {

if (j > 0) {

char\*\* text\_big = realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

count\_first++;

count\_end++;

}

text[j] = new\_str;

j++;

}

else {

count\_first++;

}

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

for (int k = 0; k < j; k++) {

printf("%s\n", text[k]);

}

printf("Количество предложений до %d и количество предложений после %d", count\_first, count\_end);

for (int l = 0; l < count\_end; l++) {

free(text[l]);

}

free(text);

return 0;

}

Программа которая по моему мнению работает хорошо

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

char end = str[i - 1];

return end;

}

int main() {

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

char\*\* text = malloc(TEXT\_SIZE \* sizeof(char\*));

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

if (check != '?') {

if (j > 0) {

char\*\* text\_big = realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

count\_first++;

count\_end++;

}

text[j] = new\_str;

j++;

}

else {

count\_first++;

}

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

for (int k = 0; k < j; k++) {

printf("%s\n", text[k]);

}

printf("Количество предложений до %d и количество предложений после %d", count\_first, count\_end);

for (int l = 0; l < count\_end; l++) {

free(text[l]);

}

free(text);

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TEXT\_SIZE 1

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

printf("(%s)\n", str);

char end = str[i - 1];

return end;

}

void processing(char\*\* text, int\* count\_first, int\* count\_end) {

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

if (check != '?') {

if (j > 0) {

char\*\* text\_big = realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

(\*count\_first)++;

(\*count\_end)++;

}

text[j] = new\_str;

printf("[%s]\n", text[j]);

j++;

}

else {

(\*count\_first)++;

}

printf("{%d, %d}", \*count\_first, \*count\_end);

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

}

void print\_res(char\*\* text, int size) {

printf("#%s#", text[0]);

printf("#%s#", text[1]);

for (int k = 0; k < size; k++) {

printf("%s\n", text[k]);

}

}

void free\_text(char\*\* text, int size) {

for (int l = 0; l < size; l++) {

free(text[l]);

}

}

int main() {

char\*\* text = malloc(TEXT\_SIZE \* sizeof(char\*));

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

processing(text, &count\_first, &count\_end);

printf("#%s#", text[0]);

printf("#%s#", text[1]);

print\_res(text, count\_end);

printf("Количество предложений до %d и количество предложений после %d", count\_first, count\_end);

free\_text(text, count\_end);

free(text);

return 0;

}

Последний

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TEXT\_SIZE 1

#define STR\_SIZE 100

char read\_str(char\* str) {

int size = STR\_SIZE;

int i = 0;

do {

if (i >= size - 2) {

char\* big\_str = (char\*) realloc(str, (size + STR\_SIZE) \* sizeof(char));

size += STR\_SIZE;

str = big\_str;

}

char symbol = getchar();

if (symbol != '\n' && symbol != '\t') {

str[i] = symbol;

i++;

}

} while (str[i - 1] != '.' && str[i - 1] != ';' && str[i - 1] != '?' && str[i - 1] != '!');

str[i] = '\0';

char end = str[i - 1];

return end;

}

void processing(char\*\* text, int\* count\_first, int\* count\_end) {

int j = 0;

do {

char\* new\_str = malloc(STR\_SIZE \* sizeof(char));

char check = read\_str(new\_str);

scanf("%\*c");

if (check != '?') {

if (j > 0) {

char\*\* text\_big = (char\*\*) realloc(text, (j + TEXT\_SIZE) \* sizeof(char\*));

text = text\_big;

(\*count\_first)++;

(\*count\_end)++;

}

text[j] = new\_str;

j++;

}

else {

(\*count\_first)++;

}

} while (strcmp(text[j - 1], "Dragon flew away!") != 0);

for (int k = 0; k < (\*count\_end)+1; k++) { // оставь если функция не зараотает

printf("%s\n", text[k]);

}

}

//void print\_res(char\*\* text, int size) { //удали если не поймешь почему не работает

// for (int k = 0; k < size; k++) {

// printf("%s\n", text[k]);

// }

//}

void free\_text(char\*\* text, int size) {

for (int l = 0; l < size; l++) {

free(text[l]);

}

}

int main() {

char\*\* text = malloc(TEXT\_SIZE \* sizeof(char\*));

int count\_first = 0; //подсчет предложений в изначальном тексте

int count\_end = 0; // подсчет предложений в конечном тексте

processing(text, &count\_first, &count\_end);

printf("Количество предложений до %d и количество предложений после %d\n", count\_first, count\_end);

//print\_res(text, count\_end);

free\_text(text, count\_end);

free(text);

return 0;

}