

# Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Дударь Юрий Мохсенович, № 5

Контакты dudaru917@gmail.com, @yuradudar

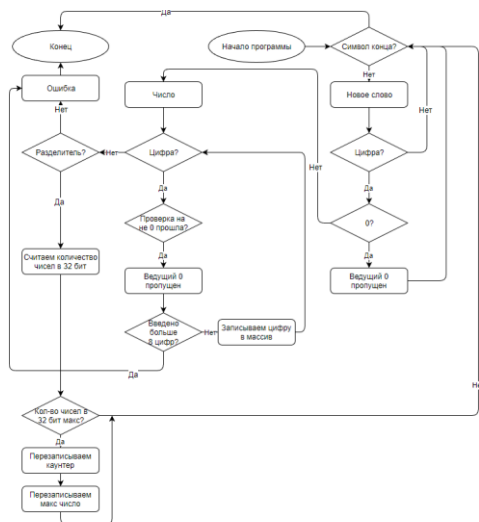
Работа выполнена: «12» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «    » \_\_\_\_\_ 20\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя

- Тема:** обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.
- Цель работы:** составить программу на Си, выполняющую анализ и обработку вводимого текста в соответствии с вариантом.
- Задание:** вариант 38 выделить все шестнадцатеричные числа, имеющие максимальное число цифр в 32 битной архитектуре.
- Оборудование (студента):**  
Процессор *Intel Core i7-12700KF @ 12x 4.5GH* с ОП 10035 Мб, НМД 102 Гб. Монитор 1920x1080
- Программное обеспечение (студента):**  
Операционная система семейства: *linux*, наименование: *mint ubuntu*, версия *18.10 cinnamon*  
интерпретатор команд: *bash* версия *4.4.19*.  
Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*
- Идея, метод, алгоритм**



## 7. Сценарий выполнения работы

Входные данные	Выходные данные	Описание результата
1ab ababaab abbab 356 634562 +	9 1ab 28 ababaab 20 abbab 10 356 23 634562 ababaab	Все правильно
0000 00 ababb	0 0 20 ababb	Все правильно
ab] +	Oops, something superfluous, I'm coming to an end	Все правильно
ababbabababa a	I accept only 8 numbers for input	Все правильно

## 8. Распечатка протокола

```
1:  /*****
2:  Если Яндекс показывает совершенно ясную погоду, а на небе собирается грозовая туча
3:  - не верь глазам своим.
4:  *****/
5:  #include <stdlib.h>
6:  #include <stdio.h>
7:  #include <assert.h>
8:  #include <stdbool.h>
9:  #include <string.h>
10: #include <ctype.h>
11:
12: typedef enum {
13:     NewWord,
14:     Number
15: } State;
16:
17: bool max_of(int a, int b){
18:     return (a > b) ? 1 : 0;
19: }
20:
21: bool is_NewWord(char symbol) {
22:     return (symbol == '\n' || symbol == '\t' || symbol == ',' || symbol == ' ') ? 1 : 0;
23: }
24:
25: bool is_Number(char symbol) {
26:     return (symbol >= '0' && symbol <= '9' || symbol >= 'a' && symbol <= 'f') ? 1 : 0;
27: }
28:
29: char append(char str1[], char str2[]){
30:     strcat(str1, str2);
31: }
32:
33: int numbers_counter(int n, char binNumber[n]){
34:     bool flag = true;
35:     int num_counter = 0;
36:     for(int i = 0; i < n*4; i++){
37:         if((binNumber[i] != '0') && (flag == true)){
38:             flag = false;
39:         }
40:         if(flag == false){
41:             num_counter++;
42:         }
43:     }
44:     return num_counter;
45: }
46:
47: char hex_to_bin(int n, char * hexNumber){
48:     char binNumber[1000] = "";
49:     char c;
50:     for (int i = 0; i < n*4; i++){
51:         c = hexNumber[i];
52:         switch (c){
53:             case '0' : append(binNumber,"0000");
54:             case '1' : append(binNumber,"0001");
55:             case '2' : append(binNumber,"0010");
56:             case '3' : append(binNumber,"0011");
57:             case '4' : append(binNumber,"0100");
58:             case '5' : append(binNumber,"0101");
59:             case '6' : append(binNumber,"0110");
60:             case '7' : append(binNumber,"0111");
61:             case '8' : append(binNumber,"1000");
62:             case '9' : append(binNumber,"1001");
63:             case 'a' : append(binNumber,"1010");
64:             case 'b' : append(binNumber,"1011");
65:             case 'c' : append(binNumber,"1100");
66:             case 'd' : append(binNumber,"1101");
67:             case 'e' : append(binNumber,"1110");
68:             case 'f' : append(binNumber,"1111");
69:         }
```

```

70:     }
71:     return numbers_counter(n,binNumber);
72: }
73:
74: void printArray (int n, char array [n]) {
75:     for (int i = 0; i < n; i++) {
76:         printf("%c", array [i]);
77:     }
78:     printf("\n");
79: }
80:
81: void test_max_of() {
82:     assert(max_of(5,4) == 1);
83:     assert(max_of(4,5) == 0);
84: }
85:
86: void test_is_NewWord() {
87:     assert(is_NewWord(' ') == 1);
88:     assert(is_NewWord('4') == 0);
89: }
90:
91: void test_is_Number() {
92:     assert(is_Number('4') == 1);
93:     assert(is_Number(']') == 0);
94: }
95:
96: void test_hex_to_bin() {
97:     assert(hex_to_bin(6,"adadad") == 24);
98:     assert(hex_to_bin(2,"cc") == 8);
99: }
100:
101: void super_tests() {
102:     test_max_of();
103:     test_is_NewWord();
104:     test_is_Number();
105:     test_hex_to_bin();
106: }
107:
108: int main () {
109:     super_tests();
110:     int max_value = 0;
111:     int ch_num = 0;
112:     State state = NewWord;
113:     char symbol, *number, *largest_number;
114:     largest_number = (char *) malloc(1);
115:     int i = 0, summ = 0, firstZero = 0;
116:     bool flag = true;
117:     while ((symbol = tolower(getchar())) != '+') {
118:         switch (state) {
119:             case NewWord:
120:                 number = (char *) malloc(1);
121:                 if (is_Number(symbol)) {
122:                     firstZero = symbol - '0';
123:                     if (firstZero == 0) {
124:                         break;
125:                     }
126:                 }
127:                 else {
128:                     flag = false;
129:                 }
130:                 state = Number;
131:                 number [i] = symbol;
132:                 i++;
133:                 number = (char *) realloc(number, i + 1);
134:                 break;
135:             case Number:
136:                 if (is_Number(symbol)) {
137:                     if (flag) {
138:                         firstZero = symbol - '0';
139:                         if (firstZero == 0) {

```

```

140:         flag = true;
141:         break;
142:     }
143:     else {
144:         flag = false;
145:     }
146: }
147: number [i] = symbol;
148: i++;
149: if(i > 8){
150:     printf("I accept only 8 numbers for input");
151:     return 0;
152: }
153: number = (char *) realloc(number, i + 1);
154: break;
155: }
156: if (is_NewWord(symbol)) {
157:     ch_num = hex_to_bin(i,number);
158:     if(max_of(ch_num,max_value)){
159:         max_value = ch_num;
160:         strcpy(largest_number, number);
161:     }
162:     printf("%d ",hex_to_bin(i,number));
163:     state = NewWord;
164:     printArray(i, number);
165:     i = 0;
166:     free(number);
167: }else{
168:     printf("Oops, something superfluous, I'm coming to an end");
169:     return 0;
170: }
171: break;
172:
173:     }
174: }
175: printArray(strlen(largest_number),largest_number);
176: return 0;
177: }

```

## 9. Дневник отладки

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	Дом	12 декабря 2022	НД	Ничего не произошло	Ничего не предпринял	Хорошо, когда ничего плохого не происходит.

## 10. Замечания автора

Интересное задание.

## 11. Выводы

Задание 38 варианта выполнил. Полученные навыки пригодятся мне в реализации следующих лабораторных работ.

Подпись студента

