

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №1
з дисципліни «Системне програмування 2»
Варіант 17

Виконав:
студент 3 курсу гр. ІО-71
Муравйов І.П.

Перевірів:
Павлов В. Г.

Київ 2019 р.

Тема: Модульне програмування в рамках базової системи проектування програм та його використання для побудови програм обробки таблиць.

Мета: Вивчення типів таблиць в системних програмах і конструкцій базової мови програмування для їх визначення. Пошук за прямою адресою. Основні типи залежностей та відношень, які реалізуються через пошук в таблицях системних програм. Лінійний та двійковий пошук.

Варіант - 17, тип вибірки – всі.

Завдання: Співпадіння максимуму початкових літер, включаючи літери латинського і слов'янського алфавіту, що співпадають за написанням.

Лістинг програми:

lab1.c:

```
1  #include <stdio.h>
2  #include <locale.h>
3  #include <string.h>
4  #include <stdbool.h>
5  #include <stdlib.h>
6  #include "utf8.h"
7  #include "lab1.h"
8
9  ch_t ch_num(char* ch)
10 {
11     int j = 0;
12     return u8_nextchar(ch, &j);
13 }
14
15 bool ch_is_same(ch_t ch1, ch_t ch2)
16 {
17     if(ch1 == ch2) {
18         return true;
19     }
20     /* cyrillic */
21     ch_t ch_ascii;
22     ch_t ch_utf;
23     if(ch1 < 255 && ch2 > 255) {
24         /* ch1 - ascii, ch2 - unicode */
25         ch_ascii = ch1;
26         ch_utf = ch2;
27     } else if(ch1 > 255 && ch2 < 255) {
28         /* ch2 - ascii, ch1 - unicode */
29         ch_ascii = ch2;
30         ch_utf = ch1;
31     } else {
32         return false;
33     }
34     if(ch_num("a") == ch_utf && ch_ascii == 'a')
35         return true;
36     if(ch_num("c") == ch_utf && ch_ascii == 'c')
37         return true;
```

```

38     if(ch_num("e") == ch_utf && ch_ascii == 'e')
39         return true;
40     if(ch_num("i") == ch_utf && ch_ascii == 'i')
41         return true;
42     if(ch_num("κ") == ch_utf && ch_ascii == 'k')
43         return true;
44     if(ch_num("n") == ch_utf && ch_ascii == 'n')
45         return true;
46     if(ch_num("o") == ch_utf && ch_ascii == 'o')
47         return true;
48     if(ch_num("z") == ch_utf && ch_ascii == 'r')
49         return true;
50     if(ch_num("x") == ch_utf && ch_ascii == 'x')
51         return true;
52     if(ch_num("y") == ch_utf && ch_ascii == 'y')
53         return true;
54     if(ch_num("A") == ch_utf && ch_ascii == 'A')
55         return true;
56     if(ch_num("B") == ch_utf && ch_ascii == 'B')
57         return true;
58     if(ch_num("C") == ch_utf && ch_ascii == 'C')
59         return true;
60     if(ch_num("E") == ch_utf && ch_ascii == 'E')
61         return true;
62     if(ch_num("H") == ch_utf && ch_ascii == 'H')
63         return true;
64     if(ch_num("K") == ch_utf && ch_ascii == 'K')
65         return true;
66     if(ch_num("M") == ch_utf && ch_ascii == 'M')
67         return true;
68     if(ch_num("O") == ch_utf && ch_ascii == 'O')
69         return true;
70     if(ch_num("P") == ch_utf && ch_ascii == 'P')
71         return true;
72     if(ch_num("X") == ch_utf && ch_ascii == 'X')
73         return true;
74     if(ch_num("V") == ch_utf && ch_ascii == 'Y')
75         return true;
76     return false;
77 }
78
79 int ch_utf_strcmp(char* str1, char* str2)
80 {
81     int i;
82     int j;
83     int count=0;
84     for(i=0, j=0; ;) {
85         ch_t ch1 = u8_nextchar(str1, &i);
86         ch_t ch2 = u8_nextchar(str2, &j);
87         if(ch1 == 0 || ch2 == 0)
88             break;
89         if(!ch_is_same(ch1, ch2)) {
90             break;
91         }
92         count++;
93     }
94     return count;
95 }
96 int ch_utf_memcmp(char* str1, char* str2)
97 {

```

```

98     int i;
99     int j;
100    for(i=0, j=0; ;) {
101        ch_t ch1 = u8_nextchar(str1, &i);
102        ch_t ch2 = u8_nextchar(str2, &j);
103        if(ch1 == 0 || ch2 == 0)
104            break;
105        if(!ch_is_same(ch1, ch2)) {
106            break;
107        }
108    }
109    u8_dec(str1, &i);
110    u8_dec(str1, &j);
111    return i;
112 }
113 char* ch_utf_diff_alloc(char*str1, char*str2)
114 {
115     int sz = ch_utf_memcmp(str1, str2);
116     char* diff = (char*)malloc(strlen(str1)+strlen(str2));
117     memcpy(diff, str1, sz);
118 }

```

lab1.h:

```

1  #ifndef MAIN_H
2  #define MAIN_H
3  #include <stdbool.h>
4  #include "utf8.h"
5  typedef unsigned int ch_t;
6  ch_t ch_num(char* ch);
7  bool ch_is_same(ch_t ch1, ch_t ch2);
8  int ch_utf_strcmp(char* str1, char* str2);
9  char* ch_utf_diff_alloc(char*str1, char*str2);
10
11
12 #endif/* MAIN_H */

```

main_lab1.c:

```

1  #include <stdio.h>
2  #include <locale.h>
3  #include <string.h>
4  #include <stdlib.h>
5  #include <assert.h>
6  #include <defs.h>
7  #include "utf8.h"
8  #include "lab1.h"
9  void tests(void)
10 {
11     char A_ascii = 'A';
12     char * A_utf = "A";
13     assert(ch_num(A_utf) == 1040);
14     assert(ch_is_same(ch_num(A_utf), A_ascii));
15     char * res = ch_utf_diff_alloc("123456", "1234556");

```

```

16     assert(strlen(res) == 5);
17 }
18 char * strs[] = { "Вакуоль", "Вакула", "Византия", "BakerStreet", "Baku", "Baobab", "binocular", "END"};
19 int
20 main() {
21     /* tests */
22     tests();
23     pr_info("locale: %s\n", setlocale(LC_ALL, ""));
24     char key[1000];
25     while(1) {
26         printf("\nkey-> ");
27         fgets(key, sizeof key, stdin);
28         int i;
29         for(i=0; ; i++) {
30             if(!strcmp(strs[i], "END"))
31                 break;
32             char* same = ch_utf_diff_alloc(strs[i], key);
33             printf(" " COLOR_RED "%s" COLOR_DEF "%s",
34                 same,
35                 &strs[i][ch_utf_memcmp(strs[i], key)]
36             );
37             free(same);
38         }
39     }
40 }

```

defs.h:

```

1  #ifndef DEFS_H
2  #define DEFS_H
3
4  #include <stdio.h>
5
6  #ifdef DEBUG_APP
7  #define pr_debug(...) do { printf("DEBUG:" __VA_ARGS__); puts(""); } while(0)
8  #else
9  #define pr_debug(...)
10 #endif
11 #define pr_info(...) do { printf(__VA_ARGS__); puts(""); } while(0)
12 #define pr_err(...) do { fprintf(stderr, "ERROR:" __VA_ARGS__); puts(""); } while(0)
13
14 #define COLOR_RED "\033[31m"
15 #define COLOR_DEF "\033[0m"
16 #endif /* DEFS_H */

```

Результати роботи програми:

```

$ ./app.elf
locale: LC_CTYPE=en_US.UTF-8;LC_NUMERIC=en_US.UTF-8;LC_TIME=en_US.UTF-8;LC_COLLATE=C;LC_MONETARY=en_US.UTF-8;LC_MESSAGES=en_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=en_US.UTF-8;LC_ADDRESS=en_US.UTF-8;LC_TELEPHONE=en_US.UTF-8;LC_MEASUREMENT=en_US.UTF-8;LC_IDENTIFICATION=en_US.UTF-8

key->
Вакуоль Вакула Византия BakerStreet Baku Baobab binocular
key->
Вакуоль Вакула Византия BakerStreet Baku Baobab binocular
key-> Ваку
Вакуоль Вакула Византия BakerStreet Baku Baobab binocular
key-> bin
Вакуоль Вакула Византия BakerStreet Baku Baobab binocular
key-> Baker
Вакуоль Вакула Византия BakerStreet Baku Baobab binocular
key-> 

```