Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

Организация эвм и систем

Лабораторная работа 3

Старались: **Шкаруба Н.Е.**
Суханова В.А.
Проверил: **Тропченко А.А.**
Группа: **P3318**

2016 г

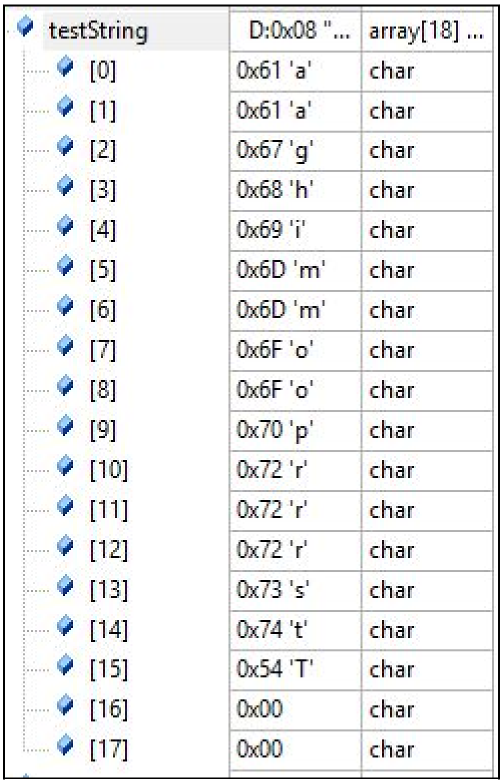
Задание:

Упорядочить текст лексикографически, в порядке возрастания ASCII-кода
"This programmator" -> "aaghimmooprrrstT"

Программа в C51:

```
#include <reg51.h>

char toLower(char letter) {
    if (letter >= 'A' && letter <= 'Z') {
        return letter - ('A' - 'a');
    } else {
      return letter;
    }
}

main() {
    char testString[] = "This programmator";
    //char code testString[] = "This programmator";
  char testStringSize = 17;
    char minChar;
    int i, pivot = 0;

    // remove all spaces
    for (i = 0; i < testStringSize; i++) {
        if (testString[i] == ' ') {
            testString[i] = testString[testStringSize - 1];
            testString[testStringSize - 1] = 0;
            testStringSize--;
        }
    }

    while(pivot != testStringSize) {
      // search for a min
        minChar = testString[pivot];

        for (i = pivot + 1; i < testStringSize; i++) {
            if (toLower(testString[i]) < toLower(minChar)) {
                minChar = testString[i];
            }
        }

        // swap min chars with chars at pivot index, increment pivot for each swap
        for (i = pivot; i < testStringSize; i++) {
            if (testString[i] == minChar) {
                // swap
                char buffer = testString[i];
                testString[i] = testString[pivot];
                testString[pivot] = buffer;
                pivot++;
            }
        }
    }
}
```

Результат выполнения:



testString	D:0x08 "...	array[18] ...
[0]	0x61 'a'	char
[1]	0x61 'a'	char
[2]	0x67 'g'	char
[3]	0x68 'h'	char
[4]	0x69 'i'	char
[5]	0x6D 'm'	char
[6]	0x6D 'm'	char
[7]	0x6F 'o'	char
[8]	0x6F 'o'	char
[9]	0x70 'p'	char
[10]	0x72 'r'	char
[11]	0x72 'r'	char
[12]	0x72 'r'	char
[13]	0x73 's'	char
[14]	0x74 't'	char
[15]	0x54 'T'	char
[16]	0x00	char
[17]	0x00	char

Файл листинга (файл с расширением *.lst)

```
C51 COMPILER V9.56.0.0   L3                                   10/23/2016
14:32:08 PAGE 1


C51 COMPILER V9.56.0.0, COMPILATION OF MODULE L3
OBJECT MODULE PLACED IN .\Objects\l3.obj
COMPILER INVOKED BY: C:\Keil_v5\C51\BIN\C51.EXE l3.c OPTIMIZE(8,SPEED)
BROWSE DEBUG OBJECTEXTEND CODE PRINT(.\Listings\l
                    -3.lst) TABS(2) OBJECT(.\Objects\l3.obj)

line level    source

   1          #include <reg51.h>
   2
   3          char toLower(char letter) {
   4    1       if (letter >= 'A' && letter <= 'Z') {
   5    2        return letter - ('A' - 'a');
   6    2       } else {
   7    2         return letter;
```

```
  8   2          }
  9   1      }
 10
 11          main() {
 12   1          char testString[] = "This programmator";
 13   1          //char code testString[] = "This programmator";
 14   1        char testStringSize = 17;
 15   1          char minChar;
 16   1          int i, pivot = 0;
 17   1
 18   1          // remove all spaces
 19   1          for (i = 0; i < testStringSize; i++) {
 20   2           if (testString[i] == ' ') {
 21   3             testString[i] = testString[testStringSize - 1];
 22   3             testString[testStringSize - 1] = 0;
 23   3             testStringSize--;
 24   3           }
 25   2          }
 26   1
 27   1          while(pivot != testStringSize) {
 28   2            // search for a min
 29   2           minChar = testString[pivot];
 30   2
 31   2           for (i = pivot + 1; i < testStringSize; i++) {
 32   3             if (toLower(testString[i]) < toLower(minChar)) {
 33   4               minChar = testString[i];
 34   4             }
 35   3           }
 36   2
 37   2           // swap min chars with chars at pivot index, increment pivot for each swap
 38   2           for (i = pivot; i < testStringSize; i++) {
 39   3             if (testString[i] == minChar) {
 40   4               // swap
 41   4               char buffer = testString[i];
 42   4               testString[i] = testString[pivot];
 43   4               testString[pivot] = buffer;
 44   4               pivot++;
 45   4             }
 46   3           }
 47   2          }
 48   1      }
*** WARNING C290 IN LINE 48 OF l3.c: missing return value
C51 COMPILER V9.56.0.0   L3                                      10/23/2016
14:32:08 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE



            ; FUNCTION _toLower (BEGIN)
                                    ; SOURCE LINE # 3
```

```
;---- Variable 'letter' assigned to Register 'R7' ----
                                  ; SOURCE LINE # 4
0000 C3              CLR     C
0001 EF              MOV     A,R7
0002 6480            XRL     A,#080H
0004 94C1            SUBB    A,#0C1H
0006 400D            JC      ?C0001
0008 D3              SETB    C
0009 EF              MOV     A,R7
000A 6480            XRL     A,#080H
000C 94DA            SUBB    A,#0DAH
000E 5005            JNC     ?C0001
                                  ; SOURCE LINE # 5
0010 EF              MOV     A,R7
0011 2420            ADD     A,#020H
0013 FF              MOV     R7,A
0014 22              RET
                                  ; SOURCE LINE # 6
0015         ?C0001:
                                  ; SOURCE LINE # 7
                                  ; SOURCE LINE # 8
                                  ; SOURCE LINE # 9
0015         ?C0002:
0015 22              RET
          ; FUNCTION _toLower (END)

          ; FUNCTION main (BEGIN)
                                  ; SOURCE LINE # 11
                                  ; SOURCE LINE # 12
0000 7800        R     MOV     R0,#LOW testString
0002 7C00        R     MOV     R4,#HIGH testString
0004 7D00              MOV     R5,#00H
0006 7BFF              MOV     R3,#0FFH
0008 7A00        R     MOV     R2,#HIGH _?ix1000
000A 7900        R     MOV     R1,#LOW _?ix1000
000C 7E00              MOV     R6,#00H
000E 7F12              MOV     R7,#012H
0010 120000      E     LCALL   ?C?COPY
                                  ; SOURCE LINE # 14
;---- Variable 'testStringSize' assigned to Register 'R1' ----
0013 7911            MOV     R1,#011H
                                  ; SOURCE LINE # 16
;---- Variable 'pivot' assigned to Register 'R4/R5' ----
0015 E4              CLR     A
0016 FD              MOV     R5,A
0017 FC              MOV     R4,A
                                  ; SOURCE LINE # 19
;---- Variable 'i' assigned to Register 'R2/R3' ----
0018 FB              MOV     R3,A
0019 FA              MOV     R2,A
```

```
001A         ?C0004:
001A E9              MOV     A,R1
001B FF              MOV     R7,A
001C 33              RLC     A
001D 95E0            SUBB    A,ACC
001F FE              MOV     R6,A
0020 C3              CLR     C
C51 COMPILER V9.56.0.0   L3                                   10/23/2016
14:32:08 PAGE 3

0021 EB              MOV     A,R3
0022 9F              SUBB    A,R7
0023 EE              MOV     A,R6
0024 6480            XRL     A,#080H
0026 F8              MOV     R0,A
0027 EA              MOV     A,R2
0028 6480            XRL     A,#080H
002A 98              SUBB    A,R0
002B 5022            JNC     ?C0008
                             ; SOURCE LINE # 20
002D 7400        R     MOV     A,#LOW testString
002F 2B              ADD     A,R3
0030 F8              MOV     R0,A
0031 E6              MOV     A,@R0
0032 B42013          CJNE    A,#020H,?C0006
                             ; SOURCE LINE # 21
0035 7400        R     MOV     A,#LOW testString+0FFFFH
0037 29              ADD     A,R1
0038 F8              MOV     R0,A
0039 E6              MOV     A,@R0
003A FF              MOV     R7,A
003B 7400        R     MOV     A,#LOW testString
003D 2B              ADD     A,R3
003E F8              MOV     R0,A
003F A607            MOV     @R0,AR7
                             ; SOURCE LINE # 22
0041 7400        R     MOV     A,#LOW testString+0FFFFH
0043 29              ADD     A,R1
0044 F8              MOV     R0,A
0045 E4              CLR     A
0046 F6              MOV     @R0,A
                             ; SOURCE LINE # 23
0047 19              DEC     R1
                             ; SOURCE LINE # 24
                             ; SOURCE LINE # 25
0048         ?C0006:
0048 0B              INC     R3
0049 BB0001          CJNE    R3,#00H,?C0019
004C 0A              INC     R2
004D         ?C0019:
```

```
004D 80CB          SJMP   ?C0004
004F        ?C0008:
                          ; SOURCE LINE # 27
004F E9           MOV    A,R1
0050 FF           MOV    R7,A
0051 33           RLC    A
0052 95E0         SUBB   A,ACC
0054 FE           MOV    R6,A
0055 EF           MOV    A,R7
0056 6D           XRL    A,R5
0057 7002         JNZ    ?C0020
0059 EE           MOV    A,R6
005A 6C           XRL    A,R4
005B        ?C0020:
005B 7003         JNZ    $ + 5H
005D 020000    R  LJMP   ?C0018
                          ; SOURCE LINE # 29
0060 7400      R  MOV    A,#LOW testString
0062 2D           ADD    A,R5
0063 F8           MOV    R0,A
0064 E6           MOV    A,@R0
0065 F500      R  MOV    minChar,A
C51 COMPILER V9.56.0.0   L3                                          10/23/2016
14:32:08 PAGE 4

                          ; SOURCE LINE # 31
0067 ED           MOV    A,R5
0068 2401         ADD    A,#01H
006A FB           MOV    R3,A
006B E4           CLR    A
006C 3C           ADDC   A,R4
006D FA           MOV    R2,A
006E        ?C0010:
006E E9           MOV    A,R1
006F FF           MOV    R7,A
0070 33           RLC    A
0071 95E0         SUBB   A,ACC
0073 FE           MOV    R6,A
0074 C3           CLR    C
0075 EB           MOV    A,R3
0076 9F           SUBB   A,R7
0077 EE           MOV    A,R6
0078 6480         XRL    A,#080H
007A F8           MOV    R0,A
007B EA           MOV    A,R2
007C 6480         XRL    A,#080H
007E 98           SUBB   A,R0
007F 5029         JNC    ?C0011
                          ; SOURCE LINE # 32
0081 7400      R  MOV    A,#LOW testString
```

```
0083 2B              ADD     A,R3
0084 F8              MOV     R0,A
0085 E6              MOV     A,@R0
0086 FF              MOV     R7,A
0087 120000      R     LCALL    _toLower
008A AE07            MOV     R6,AR7
008C AF00        R     MOV     R7,minChar
008E 120000      R     LCALL    _toLower
0091 C3              CLR     C
0092 EF              MOV     A,R7
0093 6480            XRL     A,#080H
0095 F8              MOV     R0,A
0096 EE              MOV     A,R6
0097 6480            XRL     A,#080H
0099 98              SUBB    A,R0
009A 5007            JNC     ?C0012
                             ; SOURCE LINE # 33
009C 7400        R     MOV     A,#LOW testString
009E 2B              ADD     A,R3
009F F8              MOV     R0,A
00A0 E6              MOV     A,@R0
00A1 F500        R     MOV     minChar,A
                             ; SOURCE LINE # 34
                             ; SOURCE LINE # 35
00A3         ?C0012:
00A3 0B              INC     R3
00A4 BB0001          CJNE    R3,#00H,?C0021
00A7 0A              INC     R2
00A8         ?C0021:
00A8 80C4            SJMP    ?C0010
00AA         ?C0011:
                             ; SOURCE LINE # 38
00AA AA04            MOV     R2,AR4
00AC AB05            MOV     R3,AR5
00AE         ?C0014:
00AE E9              MOV     A,R1
00AF FF              MOV     R7,A
C51 COMPILER V9.56.0.0   L3                                            10/23/2016
14:32:08 PAGE 5

00B0 33              RLC     A
00B1 95E0            SUBB    A,ACC
00B3 FE              MOV     R6,A
00B4 C3              CLR     C
00B5 EB              MOV     A,R3
00B6 9F              SUBB    A,R7
00B7 EE              MOV     A,R6
00B8 6480            XRL     A,#080H
00BA F8              MOV     R0,A
00BB EA              MOV     A,R2
```

```
00BC 6480          XRL    A,#080H
00BE 98            SUBB   A,R0
00BF 508E          JNC    ?C0008
                          ; SOURCE LINE # 39
00C1 7400      R   MOV    A,#LOW testString
00C3 2B            ADD    A,R3
00C4 F8            MOV    R0,A
00C5 E6            MOV    A,@R0
00C6 FF            MOV    R7,A
00C7 B50019    R   CJNE   A,minChar,?C0016
                          ; SOURCE LINE # 41
00CA F500      R   MOV    buffer,A
                          ; SOURCE LINE # 42
00CC 7400      R   MOV    A,#LOW testString
00CE 2D            ADD    A,R5
00CF F8            MOV    R0,A
00D0 E6            MOV    A,@R0
00D1 FF            MOV    R7,A
00D2 7400      R   MOV    A,#LOW testString
00D4 2B            ADD    A,R3
00D5 F8            MOV    R0,A
00D6 A607          MOV    @R0,AR7
                          ; SOURCE LINE # 43
00D8 7400      R   MOV    A,#LOW testString
00DA 2D            ADD    A,R5
00DB F8            MOV    R0,A
00DC A600      R   MOV    @R0,buffer
                          ; SOURCE LINE # 44
00DE 0D            INC    R5
00DF BD0001        CJNE   R5,#00H,?C0022
00E2 0C            INC    R4
00E3      ?C0022:
                          ; SOURCE LINE # 45
                          ; SOURCE LINE # 46
00E3      ?C0016:
00E3 0B            INC    R3
00E4 BB0001        CJNE   R3,#00H,?C0023
00E7 0A            INC    R2
00E8      ?C0023:
00E8 80C4          SJMP   ?C0014
                          ; SOURCE LINE # 47
                          ; SOURCE LINE # 48
00EA      ?C0018:
00EA 22            RET
     ; FUNCTION main (END)



MODULE INFORMATION:   STATIC OVERLAYABLE
  CODE SIZE        =    257    ----
  CONSTANT SIZE    =     18    ----
  XDATA SIZE       =   ----    ----
C51 COMPILER V9.56.0.0   L3                                              10/23/2016
14:32:08 PAGE 6

  PDATA SIZE       =   ----    ----
  DATA SIZE        =   ----     20
  IDATA SIZE       =   ----    ----
  BIT SIZE         =   ----    ----
END OF MODULE INFORMATION.


C51 COMPILATION COMPLETE.  1 WARNING(S),  0 ERROR(S)
```

Программа в A51:

```
; char toLower(char c)
CLR    C
MOV    A,R7
XRL    A,#080H
SUBB   A,#0C1H
JC     ?C0001
SETB   C
MOV    A,R7
XRL    A,#080H
SUBB   A,#0DAH
JNC    ?C0001
MOV    A,R7
ADD    A,#020H
MOV    R7,A
RET
?C0001:
?C0002:
ET

; Main
MOV    R0,#LOW testString
MOV    R4,#HIGH testString
MOV    R5,#00H
MOV    R3,#0FFH
MOV    R2,#HIGH _?ix1000
MOV    R1,#LOW _?ix1000
MOV    R6,#00H
MOV    R7,#012H
LCALL   ?C?COPY
MOV    R1,#011H
CLR    A
```

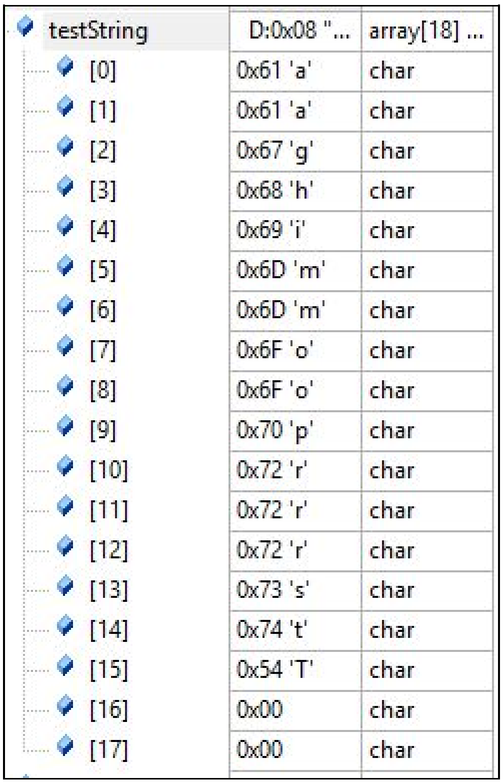
```
MOV   R5,A
MOV   R4,A
MOV   R3,A
MOV   R2,A

MOV   A,R1
MOV   R7,A
RLC   A
SUBB  A,ACC
MOV   R6,A
CLR   C
MOV   A,R3
SUBB  A,R7
MOV   A,R6
XRL   A,#080H
MOV   R0,A
MOV   A,R2
XRL   A,#080H
SUBB  A,R0
JNC   ?C0008
MOV   A,#LOW testString
ADD   A,R3
MOV   R0,A
MOV   A,@R0
CJNE  A,#020H,?C0006
MOV   A,#LOW testString+0FFFFH
ADD   A,R1
MOV   R0,A
MOV   A,@R0
MOV   R7,A
MOV   A,#LOW testString
ADD   A,R3
MOV   R0,A
MOV   @R0,AR7
MOV   A,#LOW testString+0FFFFH
ADD   A,R1
MOV   R0,A
CLR   A
MOV   @R0,A
DEC   R1

?C0006:
INC   R3
CJNE  R3,#00H,?C0019
INC   R2

?C0019:
SJMP  ?C0004

?C0008:
```

```
MOV   A,R1
MOV   R7,A
RLC   A
SUBB  A,ACC
MOV   R6,A
MOV   A,R7
XRL   A,R5
JNZ   ?C0020
MOV   A,R6
XRL   A,R4

?C0020:
NZ    $ + 5H
JMP   ?C0018
MOV   A,#LOW testString
ADD   A,R5
MOV   R0,A
MOV   A,@R0
MOV   minChar,A
MOV   A,R5
ADD   A,#01H
MOV   R3,A
CLR   A
ADDC  A,R4
MOV   R2,A
?C0010:
MOV   A,R1
MOV   R7,A
RLC   A
SUBB  A,ACC
MOV   R6,A
CLR   C
MOV   A,R3
SUBB  A,R7
MOV   A,R6
XRL   A,#080H
MOV   R0,A
MOV   A,R2
XRL   A,#080H
SUBB  A,R0
JNC   ?C0011
MOV   A,#LOW testString
ADD   A,R3
MOV   R0,A
MOV   A,@R0
MOV   R7,A
LCALL   _toLower
MOV   R6,AR7
MOV   R7,minChar
LCALL   _toLower
```

```
CLR     C
MOV     A,R7
XRL     A,#080H
MOV     R0,A
MOV     A,R6
XRL     A,#080H
SUBB    A,R0
JNC     ?C0012
MOV     A,#LOW testString
ADD     A,R3
MOV     R0,A
MOV     A,@R0
MOV     minChar,A

?C0012:
INC     R3
CJNE    R3, #00H, ?C0021
INC     R2

?C0021:
SJMP    ?C0010

?C0011:
MOV     R2,AR4
MOV     R3,AR5

?C0014:
MOV     A,R1
MOV     R7,A
RLC     A
SUBB    A,ACC
MOV     R6,A
CLR     C
MOV     A,R3
SUBB    A,R7
MOV     A,R6
XRL     A,#080H
MOV     R0,A
MOV     A,R2
XRL     A,#080H
SUBB    A,R0
JNC     ?C0008
MOV     A,#LOW testString
ADD     A,R3
MOV     R0,A
MOV     A,@R0
MOV     R7,A
CJNE    A,minChar,?C0016
MOV     buffer,A
MOV     A,#LOW testString
```

```
ADD    A,R5
MOV    R0,A
MOV    A,@R0
MOV    R7,A
MOV    A,#LOW testString
ADD    A,R3
MOV    R0,A
MOV    @R0,AR7
MOV    A,#LOW testString
ADD    A,R5
MOV    R0,A
MOV    @R0,buffer
INC    R5
CJNE   R5,#00H,?C0022
INC    R4

?C0022:
?C0016:
INC    R3
CJNE   R3,#00H,?C0023
INC    R2

?C0023:
SJMP   ?C0014

?C0018:
RET
```

Результат выполнения:



testString	D:0x08 "...	array[18] ...
[0]	0x61 'a'	char
[1]	0x61 'a'	char
[2]	0x67 'g'	char
[3]	0x68 'h'	char
[4]	0x69 'i'	char
[5]	0x6D 'm'	char
[6]	0x6D 'm'	char
[7]	0x6F 'o'	char
[8]	0x6F 'o'	char
[9]	0x70 'p'	char
[10]	0x72 'r'	char
[11]	0x72 'r'	char
[12]	0x72 'r'	char
[13]	0x73 's'	char
[14]	0x74 't'	char
[15]	0x54 'T'	char
[16]	0x00	char
[17]	0x00	char

Вывод:

Мы упорядочили текст лексикографически, в порядке возрастания ASCII- кода.