

Домашнее задание по архитектурам вычислительных систем  
Вариант 15

Михайлова Ксения Дмитриевна, БПИ192, ФКН НИУ ВШЭ

2 модуль, 2020

## Задание:

Разработать программу, вычисляющую минимальный и максимальный (по числовому значению) символы в заданной ASCII-строке

## Решение:

Для реализации алгоритма поиска минимума и максимума был использован стек.

## Вход:

Используются регистры `ecx`, `edx` для минимума и максимума соответственно.

## Цикл:

При вызове функции запоминаем текущий элемент. Изначально, в регистрах, хранящих минимальное и максимальное значения лежат максимально возможное и минимально возможные значения соответственно.

Если текущее значение — ноль-символ, то строка закончилась, и мы переходим в соответствующий блок кода, иначе переходим к проверкам.

Если символ больше минимума, то минимум не обновляется и мы переходим к следующей проверке, иначе минимум обновляется. Далее проверяем, меньше ли элемент максимума. Если меньше, то максимум не обновляется, иначе обновляем максимум.

Переходим к следующему символу и переходим в новую итерацию.

## Выход:

В `eax` записывается `ecx`, в `ebx` — `edx`.

## Организация входных данных:

Запуск программы должен производиться через консоль. Для запуска с использованием стандартной строки необходимо перейти в директорию, где расположен `.exe` файл и ввести команду: `project.exe`. Для запуска с использованием строки, заданной пользователем, необходимо перейти в директорию, где расположен `.exe` файл и ввести команду: `project.exe <строка пользователя>`.

Программа поддерживает команды: `«-h»`, `«-?»` — help, exit process with code 0.

## Организация выходных данных:

Вызывается функция вывода `sprintf`, в параметры передается соответственно: входная строка, минимум, максимум. Строка форматирует данные в виде:

«String: «<входная строка>»

min: <минимальный символ> (code: <ASCII-код минимального символа>)

max: <максимальный символ> (code: <ASCII-код максимального символа>)»

## Особенности:

Помимо латиницы, программа способна обрабатывать кириллицу.

## Примеры работы:

```
Microsoft windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>fasm project.asm project.exe
flat assembler version 1.73.25 (1048576 kilobytes memory)
4 passes, 10240 bytes.

C:\Users\Ksenia\Desktop\asm_project>project.exe

C:\Users\Ksenia\Desktop\asm_project>
```

Output

String: "Remember, you can enter any sybmols from ASCII, even Cyrillic letters!"

min: ! (code: 33)  
max: y (code: 121)

OK

```
Microsoft windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>fasm project.asm project.exe
flat assembler version 1.73.25 (1048576 kilobytes memory)
4 passes, 10240 bytes.

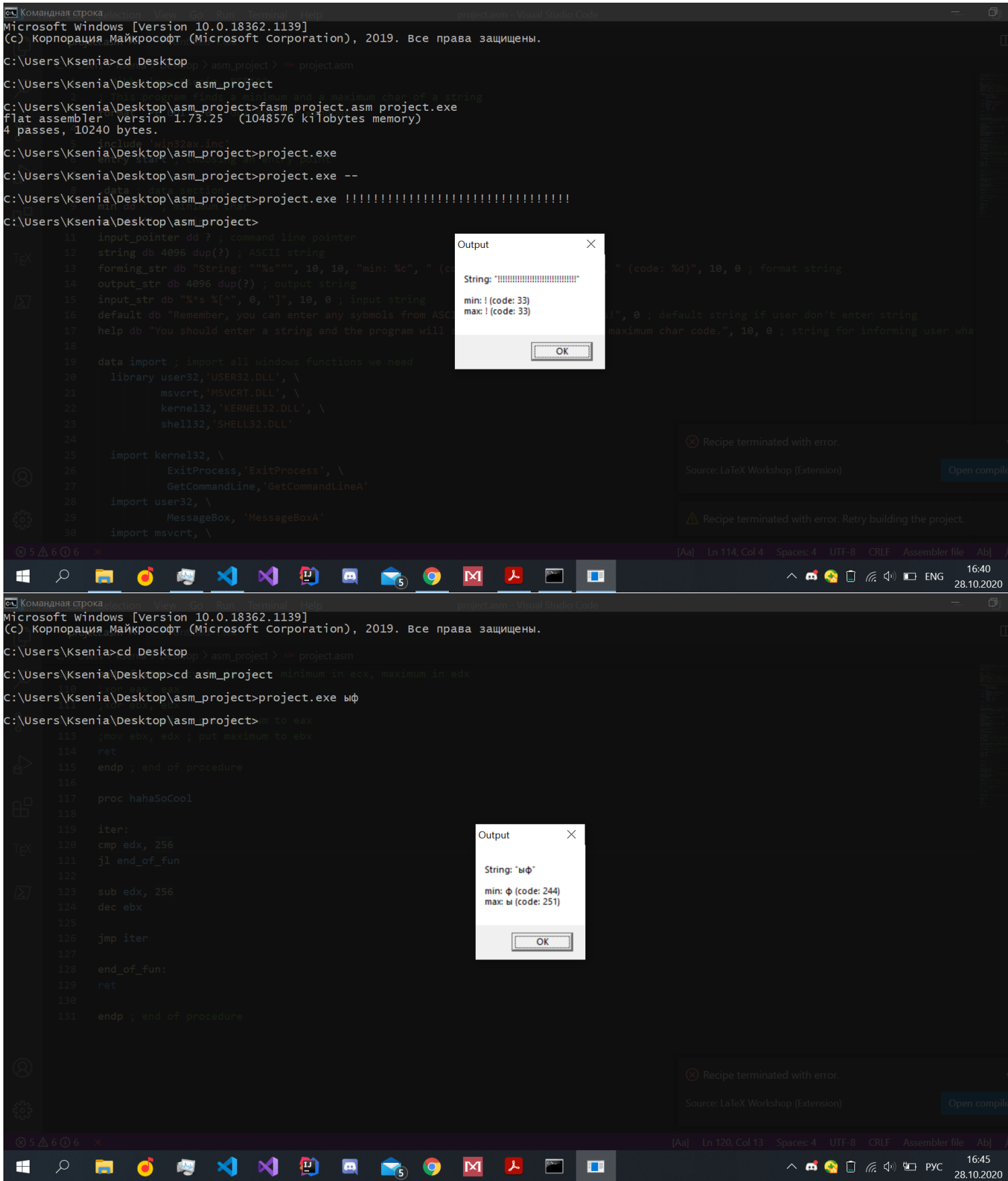
C:\Users\Ksenia\Desktop\asm_project>project.exe
C:\Users\Ksenia\Desktop\asm_project>project.exe --
C:\Users\Ksenia\Desktop\asm_project>
```

Output

String: "--"

min: - (code: 45)  
max: - (code: 45)

OK



```
Командная строка - project.asm - Visual Studio Code
Microsoft Windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>project.exe @/
C:\Users\Ksenia\Desktop\asm_project>
103 mov edx, eax ; updating maximum
104 do_not_update_max:
105
106 inc esi ; moving to the next char
107 jmp iteration ; checking of a next char
108
109 end_of_str: ; cycle is over, minimum in ecx, maximum in edx
110 ret
111 endp ; end of procedure
112
113 proc hahaSoCool
114
115 iter:
116 cmp edx, 256
117 jl end_of_fun
118
119 sub edx, 256
120 dec ebx
121
122 jmp iter
123
124 end_of_fun:
125 ret
126
127 endp ; end of procedure
```

Output

String: '@/'

min: / (code: 47)

max: @ (code: 64)

OK

```
Командная строка - project.asm - Visual Studio Code
Microsoft Windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>project.exe -h
C:\Users\Ksenia\Desktop\asm_project>
103 mov edx, eax ; updating maximum
104 do_not_update_max:
105
106 inc esi ; moving to the next char
107 jmp iteration ; checking of a next char
108
109 end_of_str: ; cycle is over, minimum in ecx, maximum in edx
110 ret
111 endp ; end of procedure
112
113 proc hahaSoCool
114
115 iter:
116 cmp edx, 256
117 jl end_of_fun
118
119 sub edx, 256
120 dec ebx
121
122 jmp iter
123
124 end_of_fun:
125 ret
126
127 endp ; end of procedure
```

Output

You should enter a string and the program will show the minimum and the maximum char code.

OK

Recipe terminated with error.

Source: LaTeX Workshop (Extension)

Open compile

Recipe terminated with error. Retry building the project.

```
Командная строка
Microsoft Windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>project.exe -h
C:\Users\Ksenia\Desktop\asm_project>project.exe -?
C:\Users\Ksenia\Desktop\asm_project>

105  inc esi ; moving to the next char
106  jmp iteration ; checking of a next char
107
108  end_of_str: ; cycle is over, minimum in eax, maximum in edx
109  ret
110
111  endp ; end of procedure
112
113  proc hahaSoCool
114
115  iter:
116  cmp edx, 256
117  jl end_of_fun
118
119  sub edx, 256
120  dec ebx
121
122  jmp iter
123
124  end_of_fun:
125  ret
126
127  endp ; end of procedure
```

Output

You should enter a string and the program will show the minimum and the maximum char code.

OK

```
Командная строка
Microsoft Windows [Version 10.0.18362.1139]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Ksenia>cd Desktop
C:\Users\Ksenia\Desktop>cd asm_project
C:\Users\Ksenia\Desktop\asm_project>project.exe -h
C:\Users\Ksenia\Desktop\asm_project>project.exe -?
C:\Users\Ksenia\Desktop\asm_project>project.exe -q
C:\Users\Ksenia\Desktop\asm_project>

105  inc esi ; moving to the next char
106  jmp iteration ; checking of a next char
107
108  end_of_str: ; cycle is over, minimum in eax, maximum in edx
109  ret
110
111  endp ; end of procedure
112
113  proc hahaSoCool
114
115  iter:
116  cmp edx, 256
117  jl end_of_fun
118
119  sub edx, 256
120  dec ebx
121
122  jmp iter
123
124  end_of_fun:
125  ret
126
127  endp ; end of procedure
```

Output

String: "-q"

min: q (code: 113)  
max: - (code: 45)

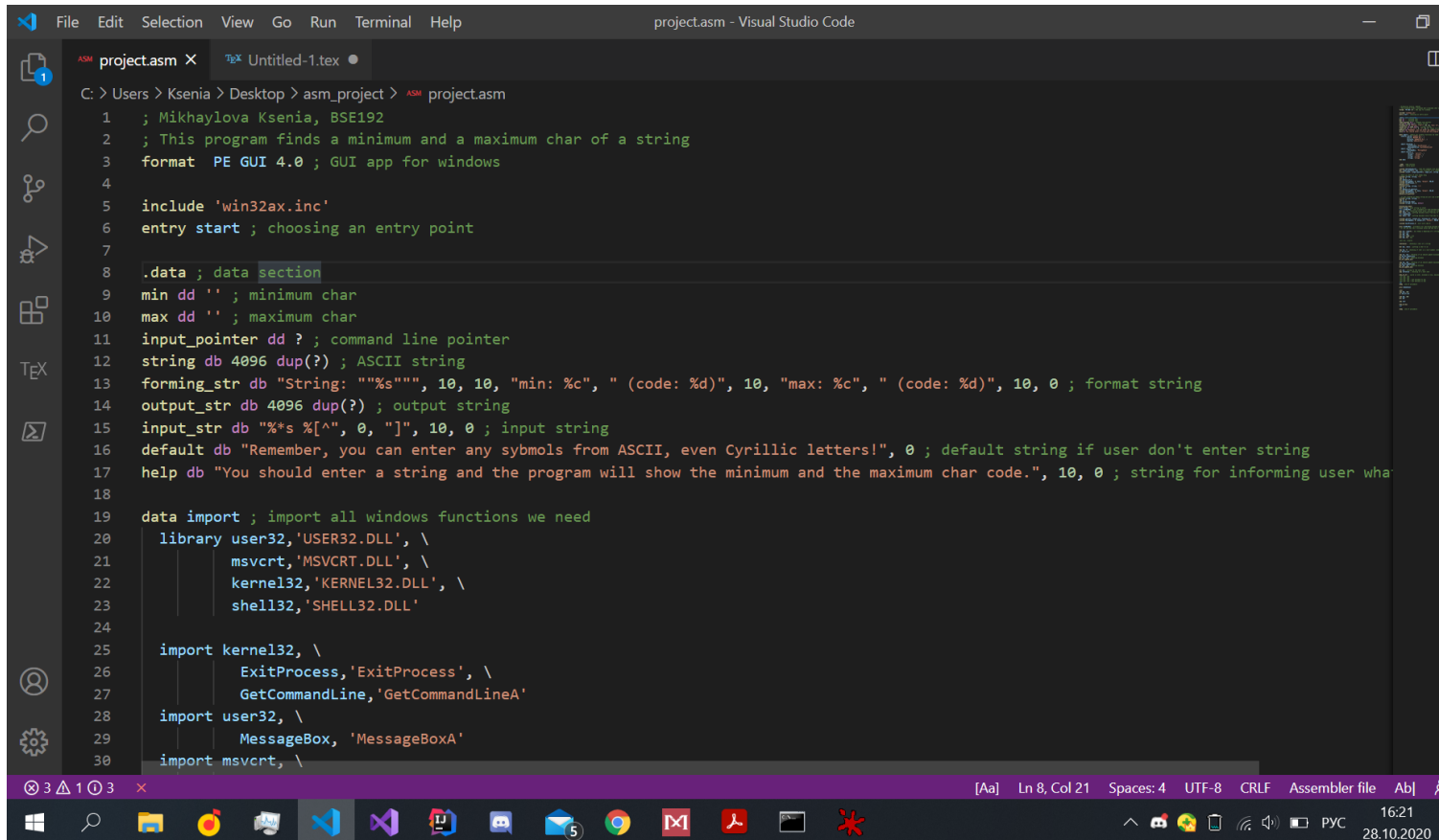
OK

## Дополнительно:

Чтобы собрать .exe файл из .asm файла необходимо, находясь в директории расположения .asm файла, ввести в консоли команду «`fasm <название1>.asm <название2>.exe`»

## Текст программы:

См. Приложение 1 (не получилось отформатировать код `fasm` в `LATEX`).



```
File Edit Selection View Go Run Terminal Help
project.asm - Visual Studio Code

ASM project.asm X Untitled-1.tex
C:\Users\Ksenia\Desktop>asm_project>ASM project.asm
1 ; Mikhailova Ksenia, BSE192
2 ; This program finds a minimum and a maximum char of a string
3 format PE GUI 4.0 ; GUI app for windows
4
5 include 'win32ax.inc'
6 entry start ; choosing an entry point
7
8 .data ; data section
9 min dd '' ; minimum char
10 max dd '' ; maximum char
11 input_pointer dd ? ; command line pointer
12 string db 4096 dup(?) ; ASCII string
13 forming_str db "String: \"%s\"", 10, 10, "min: %c", " (code: %d)", 10, "max: %c", " (code: %d)", 10, 0 ; format string
14 output_str db 4096 dup(?) ; output string
15 input_str db "%s %[^", 0, "]", 10, 0 ; input string
16 default db "Remember, you can enter any sybmols from ASCII, even Cyrillic letters!", 0 ; default string if user don't enter string
17 help db "You should enter a string and the program will show the minimum and the maximum char code.", 10, 0 ; string for informing user wha
18
19 data import ; import all windows functions we need
20 library user32,'USER32.DLL', \
21         msvcrt,'MSVCRT.DLL', \
22         kernel32,'KERNEL32.DLL', \
23         shell32,'SHELL32.DLL'
24
25 import kernel32, \
26         ExitProcess,'ExitProcess', \
27         GetCommandLine,'GetCommandLineA'
28 import user32, \
29         MessageBox, 'MessageBoxA'
30 import msvcrt, \
```

```
project.asm X  Untitled-1.tex
C:\Users\Ksenia\Desktop>asm_project>ASM project.asm
30  import msvcrt, \
31      sprintf, 'sprintf', \
32      sscanf, 'sscanf', \
33      strcmp, 'strcmp', \
34      strcpy, 'strcpy'
35  end data
36
37
38  .code ; code section
39  start: ; entry point
40
41  cinvoke GetCommandLine ; read the command line param
42  mov [input_pointer], eax ; put a pointer of a command line param from eax
43  cinvoke sscanf, [input_pointer], input_str, string ; reading input string
44
45  ; there we check if user needs help
46  cinvoke strcmp, string, "-h"
47  cmp al, 0
48  jnz absence_of_h
49  cinvoke MessageBox, 0, help, "Output", MB_OK
50  cinvoke ExitProcess, 0
51  absence_of_h:
52  cinvoke strcmp, string, "-?"
53  cmp al, 0
54  jnz absence_of_question
55  cinvoke MessageBox, 0, help, "Output", MB_OK
56  cinvoke ExitProcess, 0
57  absence_of_question:
58
59  ; if user entered an empty string we will use a default string instead
60  cinvoke strcmp, string, ""
61  cmp al, 0
62  jnz processing_input
63  cinvoke strcpy, string, default
64
65  processing_input:
66  push string ; push string to stack
67  call findMinMax ; call procedure to find minimum and maximum
68  add esp, 16 ; restoring stack after calling our procedure
69  mov [min], ecx ; putting minimum value from eax to min
70  call hahaSoCool
71  mov [max], edx ; putting maximum value from ebx to max
72
73  cinvoke sprintf, output_str, forming_str, string, [min], [min], [max], [max] ; forming an output string
74  cinvoke MessageBox, 0, output_str, "Output", MB_OK ; output
75
76  cinvoke ExitProcess, 0 ; exit with code 0
77
78  proc findMinMax ; procedure for searching minimum and maximum char of a string
79  ; at the end eax has a minimum value and ebx has a maximum value
80
81  mov esi, [esp+4] ; esi keeps a beginning of a string
82  xor ecx, ecx
83  xor edx, edx
84  mov ecx, 255d ; min
85  mov edx, 0d ; max
86
87  ;mov esi, [esp+4]
88
```



```
File Edit Selection View Go Run Terminal Help project.asm - Visual Studio Code
ASM project.asm X Untitled-1.tex
C:\Users> Ksenia > Desktop > asm_project > ASM project.asm
87 ;mov eax, [esi]
88
89 iteration: ; checking a char of a string
90
91 mov eax, [esi] ; putting a char to al
92
93 cmp eax, 0 ; checking if char is a null-symbol (end of string)
94 jz end_of_str
95
96 cmp ecx, eax ; checking if we should update minimum
97 jb do_not_update_min
98 mov ecx, eax ; updating minimum
99 do_not_update_min:
100
101 cmp eax, edx ; checking if we should update maximum
102 jb do_not_update_max
103 mov edx, eax ; updating maximum
104 do_not_update_max:
105
106 inc esi ; moving to the next char
107 jmp iteration ; checking of a next char
108
109 end_of_str: ; cycle is over, minimum in ecx, maximum in edx
110 ;xor eax, eax
111 ;xor ebx, ebx
112 ;mov eax, edx ; put minimum to eax
113 ;mov ebx, edx ; put maximum to ebx
114 ret
115 endp ; end of procedure
116
117 proc hahaSoCool
118
119 iter:
120 cmp edx, 256
121 jl end_of_fun
122
123 sub edx, 256
124 dec ebx
125
126 jmp iter
127
128 end_of_fun:
129 ret
130
131 endp ; end of procedure
```

[Aa] Ln 96, Col 52 Spaces: 4 UTF-8 CRLF Assembler file Abj 16:21 28.10.2020