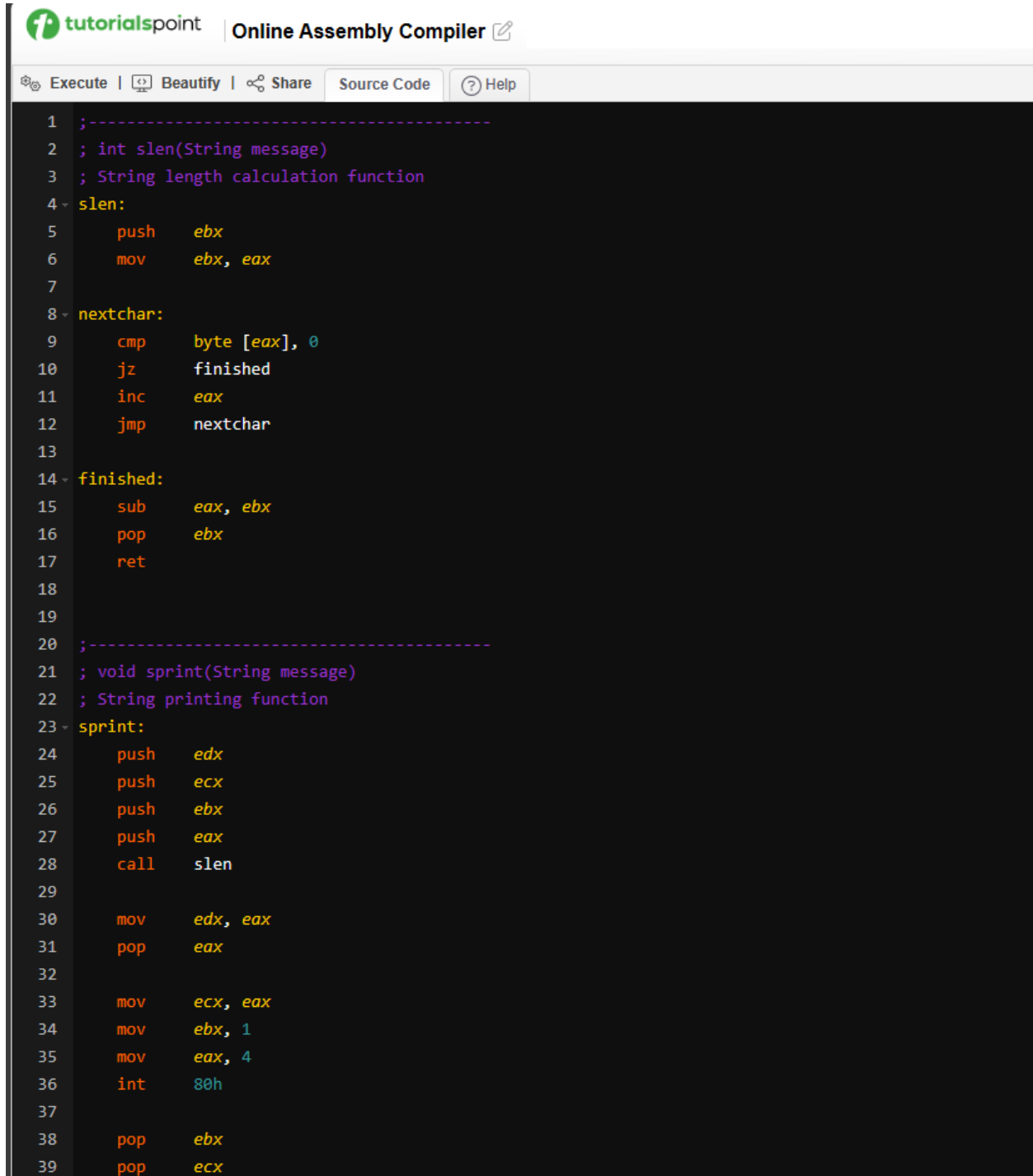


Keene Keannu Kurt C. De Jesus

BSCS3-1



The screenshot displays the 'Online Assembly Compiler' interface from 'tutorialspoint'. The interface includes a header with the logo and name, and a navigation bar with buttons for 'Execute', 'Beautify', 'Share', 'Source Code', and 'Help'. The main area contains assembly code for two functions: `slen` and `sprint`. The `slen` function calculates the length of a string by pushing `ebx`, moving `eax` to `ebx`, and then looping through the string until a null terminator is found. The `sprint` function pushes `edx`, `ecx`, `ebx`, and `eax`, calls `slen`, and then moves the return value back to `edx` and `ecx` before popping `ebx` and `ecx`.

```
1  ;-----  
2  ; int slen(String message)  
3  ; String length calculation function  
4  slen:  
5      push    ebx  
6      mov     ebx, eax  
7  
8  nextchar:  
9      cmp     byte [eax], 0  
10     jz      finished  
11     inc     eax  
12     jmp     nextchar  
13  
14 finished:  
15     sub     eax, ebx  
16     pop     ebx  
17     ret  
18  
19  
20 ;-----  
21 ; void sprint(String message)  
22 ; String printing function  
23 sprint:  
24     push    edx  
25     push    ecx  
26     push    ebx  
27     push    eax  
28     call    slen  
29  
30     mov     edx, eax  
31     pop     eax  
32  
33     mov     ecx, eax  
34     mov     ebx, 1  
35     mov     eax, 4  
36     int     80h  
37  
38     pop     ebx  
39     pop     ecx
```

```

40     pop     edx
41     ret
42
43
44 ;-----
45 ; void sprintLF(String message)
46 ; String printing with line feed function
47 ~ sprintLF:
48     call    sprint
49
50     push    eax        ; push eax onto the stack to preserve it while we use the eax register in
                        ; this function
51     mov     eax, 0Ah    ; move 0Ah into eax - 0Ah is the ascii character for a linefeed
52     push    eax        ; push the linefeed onto the stack so we can get the address
53     mov     eax, esp    ; move the address of the current stack pointer into eax for sprint
54     call    sprint     ; call our sprint function
55     pop     eax        ; remove our linefeed character from the stack
56     pop     eax        ; restore the original value of eax before our function was called
57     ret             ; return to our program
58
59
60 ;-----
61 ; void exit()
62 ; Exit program and restore resources
63 ~ quit:
64     mov     ebx, 0
65     mov     eax, 1
66     int     80h
67     ret
68
69 SECTION .data
70 msg1     db      'Hello, brave new world!', 0h        ; NOTE we have removed the line feed
                        ; character 0Ah
71 msg2     db      'This is how we recycle in NASM.', 0h ; NOTE we have removed the line feed
                        ; character 0Ah
72
73 SECTION .text
74 global _start
75

```

```

70 msg2     db      'This is how we recycle in NASM.', 0h ; NOTE we have removed the line feed
                        ; character 0Ah
71
72
73 SECTION .text
74 global _start
75
76 ~ _start:
77
78     mov     eax, msg1
79     call    sprintLF    ; NOTE we are calling our new print with linefeed function
80
81     mov     eax, msg2
82     call    sprintLF    ; NOTE we are calling our new print with linefeed function
83
84     call    quit

```

FacebookOnline Assembly CompilerLPUL_2324,1T_4046 Multitem - / xloa-todo - Engraving Calculator

https://www.tutorialspoint.com/compile_asm_online.php

ProjectEditSettingLogIn

Execute | Beautify | Share | Source Code | Help

Terminal

```
1 ;-----
2 ; int strlen(String message)
3 ; String length calculation function
4 strlen:
5     push    ebx
6     mov     ebx, eax
7
8 nextchar:
9     cmp     byte [ebx], 0
10    jz      finished
11    inc     eax
12    jmp     nextchar
13
14 finished:
15    sub     eax, ebx
16    pop     ebx
17    ret
18
19
20 ;-----
21 ; void sprintf(String message)
22 ; String printing function
23 sprintf:
24     push    edx
25     push    ecx
26     push    ebx
27     push    eax
28     call    strlen
29
30     mov     edx, eax
31     pop     eax
32
33     mov     ecx, eax
34     mov     ebx, 1
35     mov     eax, 4
36     int     0x80
37
38     pop     ebx
39     pop     ecx
```

```
Error while connecting with the serverbash: /home/webmaster/.bashrc: Permission denied
webmaster@dedec7b2669:/home/cg/root/6544dd581bb6$ webmaster@dedec7b2669:/home/cg/root/6544dd581bb6$ nasm -f elf *.asm; ld -m
elf
_1386 -s -o demo *.o && demo

Hello, brave new world!
This is how we recycle in NASM.
webmaster@dedec7b2669:/home/cg/root/6544dd581bb6$
```