Comp 3350: Computer Organization & Assembly Language
HW # 10: Theme: Strings and Arrays

(All main questions carry equal weight. Credit awarded to only those answers for which work has been shown.)

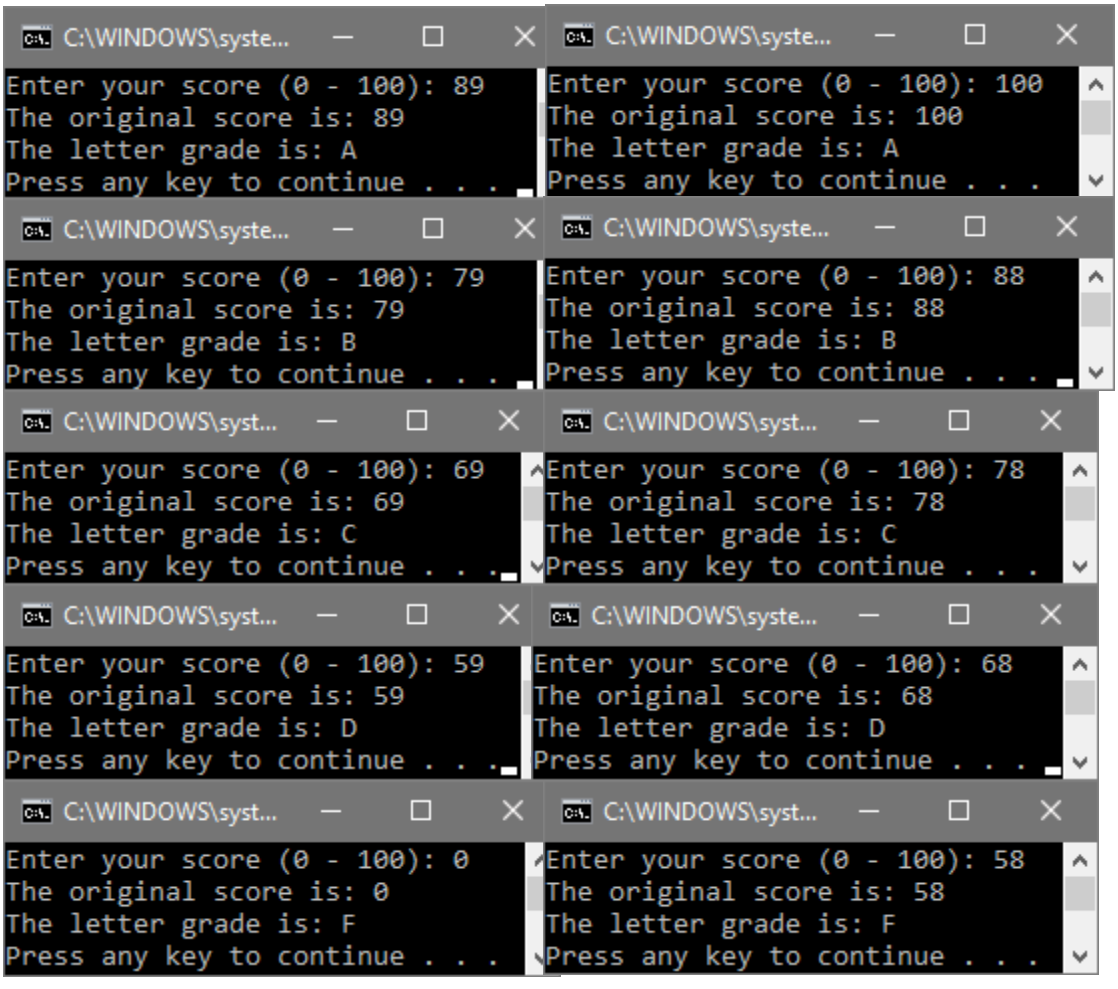
---

1. [Case Table] Write a program that asks the user to enter a score. It then adds 1 to the score, creating a final score and prints the letter grade based on the final score (see table below). The program should display your original score, final score as well as the letter grade. You should reference the section of the text that discusses Table Driven Selection. Use the following data as a guide for letter grade and score range association:

Score Range	Letter Grade
89 – 100	A
79 – 88	B
69 - 78	C
59 – 68	D
0 - 58	F

Please embed your code into your homework submission along with a screen shot post execution.

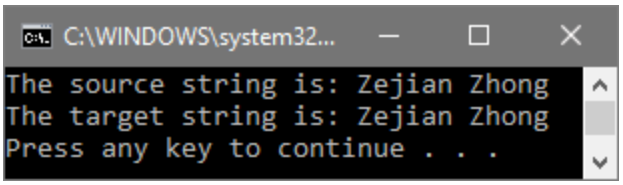
```
INCLUDE Irvine32.inc
.data
score BYTE "Enter your score (0 - 100): ", 0
msg1 BYTE "The original score is: ", 0
msg2 BYTE "The letter grade is: ", 0
out1 BYTE "A", 0
out2 BYTE "B", 0
out3 BYTE "C", 0
out4 BYTE "D", 0
out5 BYTE "F", 0
.code
main PROC
mov edx, OFFSET score
call WriteString
call ReadDec
mov edx, OFFSET out1
cmp eax, 89
jae LBX
mov edx, OFFSET out2
cmp eax, 79
jae LBX
mov edx, OFFSET out3
cmp eax, 69
jae LBX
mov edx, OFFSET out4
cmp eax, 59
jae LBX
mov edx, OFFSET out5
cmp eax, 0
jae LBX
LBX:
push edx
mov edx, OFFSET msg1
call WriteString
```



```
call WriteDec
call Crlf
pop edx
push edx
mov edx, OFFSET msg2
call WriteString
pop edx
call WriteString
call Crlf
exit
main ENDP
END main
```

2. [Strings] Write a program that copies bytes from source to target. You must use string instructions to accomplish the job. Declare the source and target locations in the data segment. The source string should be your name.

```
INCLUDE Irvine32.inc
.data
str1 BYTE "Zejian Zhong", 0
str2 BYTE 20 DUP(?)
msg1 BYTE "The source string is: ", 0
msg2 BYTE "The target string is: ", 0
.code
main PROC
mov edx, OFFSET msg1
call WriteString
mov edx, OFFSET str1
call WriteString
call Crlf
mov edx, OFFSET msg2
call WriteString
cld
mov ecx, LENGTHOF str1
mov esi, OFFSET str1
mov edi, OFFSET str2
rep movsb
mov edx, OFFSET str2
call WriteString
call Crlf
    EXIT
main ENDP
END main
```

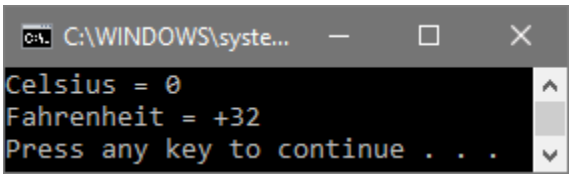


3. [General Programming] Write a program that converts the temperature in Celsius to *F* in Fahrenheit using *F = C*9/5 + 32*. For ease of programming you can display the result in fractions, i.e. *20 1/9* (no need to use floats, just display the quotient, the slash character and the digit 9). Show the runs for freezing, boiling point, room temperature and human body temperature. Provide screen shots of the runs along with your program.

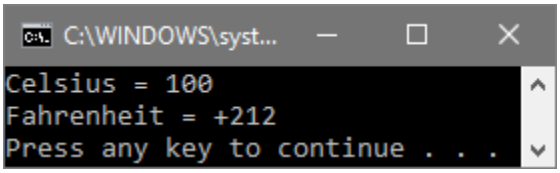
```
INCLUDE Irvine32.inc
.data
cel BYTE "Celsius = ", 0
fah BYTE "Fahrenheit = ", 0
denominator BYTE "/5", 0
space BYTE " ", 0
```

```
.code
main proc
mov edx, OFFSET cel
call WriteString
call ReadInt
push eax
call converter
call Crlf
exit
main ENDP

converter proc
push ebp
mov ebp, esp
mov eax, [ebp+8]
imul eax, 9
mov ebx, +5
cdq
idiv ebx
add eax, 32
push edx
mov edx, OFFSET fah
call WriteString
pop edx
call WriteInt
push edx
mov edx, OFFSET space
call WriteString
pop edx
push eax
mov eax, edx
test eax,eax
jns NonNeg
theNeg:
neg eax
NonNeg:
cmp eax, 0
je fract
call WriteDec
pop eax
push edx
mov edx, OFFSET denominator
call WriteString
pop edx
jmp endexit
fract:
pop eax
endexit:
pop ebp
ret 4
converter ENDP
END main
```

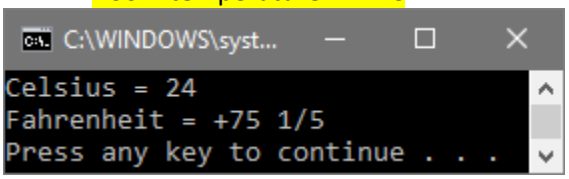
Freezing point = 0°C



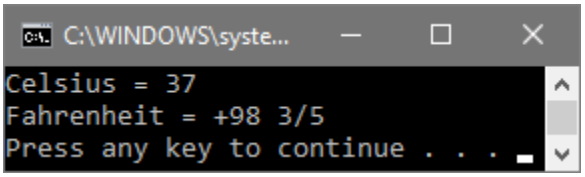
Boiling point = 100°C



Room temperature = 24°C



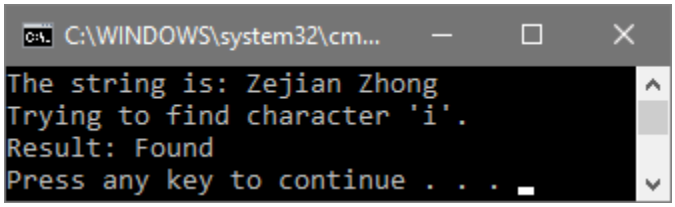
Body temperature = 37°C



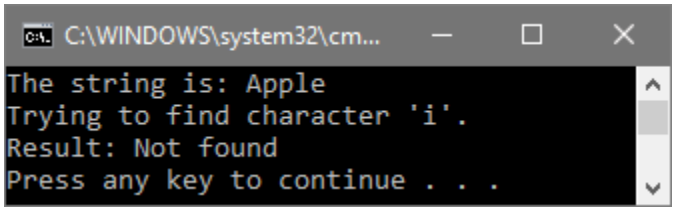
4. [Strings] Write a program that searches for a character in a string. You should set the EDI pointer to point to the character found. Test the program thoroughly using various strings, including your name. Provide screen shots of the runs along with your program. You must use string instructions in your program.

```
INCLUDE Irvine32.inc
.data
MAX = 100
str0 BYTE "The string is: ", 0
str1 BYTE MAX+1 DUP(?)
str2 BYTE "Trying to find character 'i'.", 0
found BYTE "Result: Found", 0
notfound BYTE "Result: Not found", 0
.code
main PROC
push edx
mov edx, OFFSET str0
call WriteString
mov edx, OFFSET str1
mov ecx, MAX
call ReadString
mov edx, OFFSET str2
call WriteString
call Crlf
mov edi, OFFSET str1
mov al, 'i'
mov ecx, LENGTHOF str1
cld
repne scasb
jnz cannotfound
dec edi
mov edx, OFFSET found
call WriteString
call Crlf
jmp quit

cannotfound:
mov edx, OFFSET notfound
call WriteString
call Crlf

quit:
exit
main ENDP
END main
```

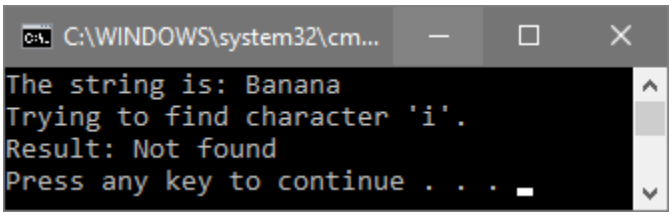
My name



Apple



Banana



iPhone X