



Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : CSE2214
Course Title : Assembly Language Programming Sessional
Assignment No : 11

Date of Performance : 16.02.22
Date of Submission : 23.02.22

Submitted To : Mr. Sajib Kumar Saha Joy & Zarin Tasnim Shejuti

Submitted By

Group : B1
Name : Kazi Atiqur Rahman
Id : 190204086
Section : B

Question No: 01

Question: Write a program that (a) lets the user input a string, (b) prints it forward and backward without punctuation and blanks on successive lines, and (c) decides whether it is a palindrome and prints the conclusion

Answer:

.MODEL SMALL

.STACK 100H

.DATA

INPUT_MSG DB "ENTER A STRING: \$"

OUTPUT_MSG_Y DB 0DH, 0AH, "IT IS A PALINDROME. \$"

OUTPUT_MSG_N DB 0DH, 0AH, "IT IS NOT A PALINDROME. \$"

INPUT_STR DB 80 DUP(0)

REVERSED_STR DB 80 DUP(0)

NEW_LINE DB 0DH, 0AH, '\$'

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

LEA DX, INPUT_MSG

MOV AH, 9

INT 21H

LEA DI, INPUT-STR

CALL READ-STR

LEA DX, NEW-LINE

MOV AH, 9

INT 21H

LEA SI, INPUT-STR

CALL DISP-STR

LEA DI, REVERSED-STR

MOV CX, BX

MOV SI, BX

DEC SI

REVERSE:

MOV AL, INPUT-STR[SI]

STOSB

DEC SI

LOOP REVERSE

LEA DX, NEWLINE

MOV AH, 9

INT 21H

LEA SI, REVERSED-STR

CALL DISP-STR

CLD

```
LEA SI, INPUT_STR
LEA DI, REVERSED_STR
MOV CX, BX
```

CHECK:

```
CMP SB
JNZ NO
LOOP CHECK
```

```
LEA DX, OUTPUT_MSGY
MOV AH, 9
INT 21H
JMP EXIT1
```

NO:

```
LEA DX, OUTPUT_MSGN
MOV AH, 9
INT 21H
```

EXIT1:

```
MOV AH, 4CH
INT 21H
MAIN ENDP
```

```
INCLUDE input_string_a.asm
INCLUDE output_string_a.asm
```

END MAIN

READ_STR PROC

PUSH AX

PUSH ~~DI~~ DI

CLD

XOR BX, BX

MOV AH, 1

INT 21H

INPUT:

CMP AL, 0DH

JE END-INPUT

CMP AL, 8H

JNE CHECK_NO

DEC DI

DEC BX

JMP READ

~~GET~~ CHECK_NO:

CMP AL, '0'

JB READ

CMP AL, ':'

JB LEGAL

CHECK_CAPITAL:

CMP AL, 'A'

JB READ

```
CMP AL, '['  
JB LEGAL  
CHECK_SMALL:  
CMP AL, 'a'  
JB READ  
CMP AL, 'z'  
JA READ  
  
LEGAL:  
STOSB  
INC BX  
  
READ:  
INT21H  
JMP INPUT  
  
END_INPUT:  
POP DI  
POP AX  
RET  
READ_STR ENDP
```

DISP_STR PROC

PUSH AX

PUSH BX

PUSH CX

PUSH DX

PUSH SI

MOV CX, BX

JC XZ EXIT

CLD

MOV AH, 2

PRINT:

LODSB

MOV DL, AL

INT 21H

LOOP PRINT

EXIT:

POP SI

POP DX

POP CX

POP BX

POP AX

RET

DISP_STR ENDP

Question No: 02

Question: Write a program that reads a string `STRING`, a decimal integer `S` that represents a position in `STRING`, a decimal integer `N` that represents the number of bytes to be removed (both integers between 0 and 80), calls `DELETE` to remove `N` bytes at position `S`, and prints the resulting string

Answer:

.MODEL SMALL

.STACK 100H

.DATA

INPUT_MSG DB "ENTER A STRING: \$"

INPUT_S DB 0DH, 0AH, "ENTER POSITION: \$"

INPUT_N DB 0DH, 0AH, "ENTER NUMBER OF BYTE: \$"

OUTPUT_MSG DB 0DH, 0AH, "RESULTED STRING: \$"

STRING DB 80 DUP(0)

NEWLINE DB 0DH, 0AH, '\$'

S DB ?

N DB ?

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

LEA DX, INPUT_MSG

MOV AH, 9

INT 21H

LEA DI, STRING

CALL READ_STR

LEA DX, INPUT_S

MOV AH, 9

INT 21H

CALL INPUT_DEC

MOV S, AL

LEA DX, INPUT_N

MOV AH, 9

INT 21H

CALL INPUT_DEC

MOV N, AL

XOR AH, AH

MOV AL, S

CMP AX, BX

JG OUTPUT

CMP N, 0

JE OUTPUT

LEA SI, STRING

CALL DELETE

OUTPUT:

LEA DX, OUTPUT_MSG

MOV AH, 9

INT 21H

LEA SI, STRING

CALL DISP_STR

MOV AH, 4CH

INT 21H

MAIN ENDP

INCLUDE input_string_b.asm

INCLUDE input_dec.asm

INCLUDE output_string_b.asm

INCLUDE delete.asm

END MAIN

input_string_b.asm

READ_STR PROC

PUSH AX

PUSH DI

CLD

XOR BX, BX

MOV AH, 1

INT 21H

INPUT:

CMP AL, 0DH

JE END_INPUT

```
CMP AL, 8H  
JNE NOT-BACK
```

```
DEC DI  
DEC BX  
JMP READ
```

```
NOT-BACK  
STOSB  
INC BX
```

```
READ:  
INT 21H  
JMP INPUT
```

```
END-INPUT:
```

```
POP DI  
POP AX
```

```
RET
```

```
READ-STR ENDP
```

input_dec.asm

```
INPUT-DEC PROC
```

```
PUSH BX  
PUSH CX  
PUSH DX
```

```
BEGIN:
```

```
XOR BX, BX
```

MOV AH, 1

INT 21H

REPEAT2:

CMP AL, '0'

JNGE NOT-DIGIT

CMP AL, '9'

JNLE NOT-DIGIT

AND AX, 000FH

PUSH AX

MOV AH, 10

MUL BX

POP BX

ADD BX, AX

MOV AH, 1

INT 21H

CMP AL, 0DH

JNE REPEAT2

MOV AX, BX

EXIT2:

POP DX

POP CX

POP BX

RET

```
NOT-DIGIT :  
MOV AH, 2  
MOV DL, 0DH  
INT 21H  
MOV DL, 0AH  
INT 21H  
JMP BEGIN
```

```
INPUT-DEC ENDP
```

delete.asm

```
DELETE PROC
```

```
PUSH BX  
PUSH SI  
PUSH DI
```

```
CWD  
MOV CX, BX  
XOR BH, BH  
MOV BL, 5
```

```
LEA SI, STRING  
LEA DI, STRING
```

```
SUB CX, BX  
ADD SI, BX  
ADD DI, BX  
XOR BH, BH  
MOV BL, N  
ADD SI, BX
```

REMOVE:

MOVSB

LOOP REMOVE

EXIT1:

POP DI

POP SI

POP BX

RET

DELETE ENDP

DISP_STR PROC

PUSH AX

PUSH BX

PUSH CX

PUSH DX

PUSH SI

MOV CX, BX

JCXZ EXIT

CLD

MOV AH, 2

PRINT:

LODSB

MOV DL, AL

INT 21H

LOOP PRINT

EXIT:

POP SI

POP DX

POP CX

POP BX

POP AX

RET

DISP_STR ENDP