	Enter a string and check whether the string is  Palisher and check whether the string is	(
(LAB PROGR	Palindrome on not.	
	· MODEL SMALL	-
	DISPLAY MACRO MSG. HLW and vom	-
	LEADX, MSCr and ab vim	
	MOV AH, O9H - ABS, It VUIS	
	INT 21H mul tai	- (
	ENDM ENDM	-
	DATA Les day kes	
	MSGI DB ODH, OAH, "ENTER STRING: 800 JUS	
	MSG2 DB ODH, OAH, " REVERSE STRING : \$" WITH	
	MSG3 DB ODH, OAH, "INPUT STRING IS PALINDROME. \$"	
	MSG4 DB DDH, OAH, "INPUT STRING IS NOT A PALINDROME STRING	2.5
	STRING DB 80H DUP(9) com song gain	
	RESTRING DB 80H DUP(?) HTO NO VOM	
	. CODE	
	START: MOV AX, @ DATA	
	Mov Ds, Ax gons gons	
	DISPLAY MSGL . DOS	a <sub>e</sub>
	; TAKE THE STRING FROM KEYBOARD CHARACTER BY	
	CHARACTER	
	MOV SI, OFFSET STRING	-
	XOR CL, CL	
	AGAIN: MOV AH, OIH	
	INT 21H	
	CMP AL, ODH	-1
	JE NEXT	
	MOV [SI], AL	
	INC SI	- 1
	INC CL	
	JMP AGAIN	

MOV DI, OFFSET RSTRING.  AGe: MOV AL, [SI]  CMP AL, [DI]		
MOV CH, CL; REVERSE THE STRING AND STORE IN RETRING  MOV DI, OFFSET RETRING  BACK: MOV AL, [SI]  MOV [DI], AL  DEC SI  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR '\$'  DISPLAY MSG2  DISPLAY RETRING  MOV SI, OFFSET STRING  MOV DI, OFFSET RETRING  AG: MOV AL, [SI]  TNC EALL		NEXT: MOV [SI] BYTE PTR '\$'; STRING INPUT OVER
MOV DI, OFFSET RSTRING  BACK: NOV AL, [SI]  MOV [DI], AL  DEC SI  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG12  DISPLAY RSTRING  NOV SI, OFFSET STRING  NOV DI, OFFSET RSTRING  AG: NOV AL, [SI]  CMP AL, [DI]  JNE FAIL  INC SI  TNC DI  DEC CX  TZ SUCCESS  TMP AG.  FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4CH  INT 21H		DEC SI
MOV DI, OFFSET RSTRING  BACK: MOV AL, [SI]  MOV [DI], AL  DEC SI  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY RSTRING  MOV SI OFFSET STRING  NOV DI, OFFSET RSTRING  AGC: MOV AL, [SI]  TNE EALL		MOV CH, CL , REVERSE THE STRING AND CTORE IN ROTAL
BACK: MOV AL, [SI]  MOV [DI], AL  DEC SI  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY RSTRING  MOV SI, OFFSET STRING  MOV DI, OFFSET RSTRING  AG: MOV AL, [SI]  TNG EAN		MOV DI, OFFSET RSTRING
MOV [DI], AL  DEC SI  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY MSG2  DISPLAY RSTRING  MOV SI, OFFSET STRING  MOV DI, OFFSET RSTRING  AGC: MOV AL, [SI]  TNE EALL	~	
DEC ST  INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY RSTRINGL  NOV SI, OFFSET STRINGL  NOV DI, OFFSET RSTRINGL  AGC: MOV AL, [SI]  TNE EALL		
INC DI  DEC CH  JNZ BACK  MOV [DI], BYTE PTR '\$'  DISPLAY MSG2  DISPLAY RSTRING  MOV SI, OFFSET STRING  MOV DI, OFFSET RSTRING  AG: MOV AL, [SI]  TNG EALL		
DEC CH  JNZ BACK  MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY RSTRINGL  MOV SI, OFFS ET STRINGL  MOV DI, OFFS ET RSTRINGL  AGC: MOV AL, [SI]  CMP AL, [DI]		
MOV [DI], BYTE PTR '\$'  DISPLAY MSG2  DISPLAY RSTRINGL  MOV SI, OFFSET STRING  MOV DI, OFFSET RSTRINGL  AG: MOV AL, [SI]  CMP AL, [DI]		
MOV [DI], BYTE PTR'\$'  DISPLAY MSG2  DISPLAY RSTRINGL  MOV SI, OFFSET STRING  NOV DI, OFFSET RSTRING  AG: MOV AL, [SI]  CMP AL, [DI]		JNZ BACK
DISPLAY MSG2  DISPLAY RSTRING  MOV SI, OFFSET STRING  MOV DI, OFFSET RSTRING  AG: MOV AL, [SI]  CMP AL, [DI]		
DISPLAY RSTRING.  MOV SI, OFFSET STRING.  MOV DI, OFFSET RSTRING.  AGe: MOV AL, [SI]  CMP AL, [DI]		
MOV SI, OFFSET STRING.  MOV DI, OFFSET RSTRING.  AGE: MOV AL, [SI]  CMP AL, [DI]		
MOV DI, OFFSET RSTRING.  AGe: MOV AL, [SI]  CMP AL, [DI]  TNE FAIL		
AGE: MOV AL, [SI]  CMP AL, [DI]  TNG FALL		
TNC EAIL		and the second s
TNC EAIL		CMP AL, [DI]
INC SI  INC DI  DEC CX  JZ SUCCESS  JNP A G2  FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4 CH  INT 21H  END		
TNC DI  DEC CX  JZ SUCCESS  JNP AG  FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4CH  INT 21H  END		INC SI
DEC CX  JZ SUCCESS  JNP AG  FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4CH  INT 21H  END		INCDI
JZ SUCCESS  JMP AG  FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4CH  INT 21H  END		
FAIL: DISPLAY MSG4  JMP FINAL  SUCCESS: DISPLAY MSG3  FINAL: MOV AH, 4CH  INT 21H  END		
FAIL: DISPLAY MSGG  JMP FINAL  SUCCESS: DISPLAY MSGGS  FINAL: MOV AH, 4CH  INT 21H  END		
SUCCESS: DISPLAY MSG13  FINAL: MOV AH, 4CH  INT 21H  END		The second secon
FINAL: MOV AH, 4CH  INT 21H  END		
INT 21H END		
END		