DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,

FACULTY OF ECE,

**Rajshahi University of Engineering & Technology, Bangladesh**

EEE - 3210– Microprocessor, Interfacing & System Design Sessional

**Sessional Assignment**

Submitted to

**Belal Hossain**

Assistant Professor

Dept. of Electrical & Electronic Engineering,

Rajshahi University of Engineering and Technology.

Submitted by

**Ashraf Al- Khalique**

Roll: 1801171

Session: 2018-2019

Dept. of Electrical & Electronic Engineering,

Rajshahi University of Engineering and Technology.

**Find prime and non- prime number**

* **Emulator Code**

|  |  |
| --- | --- |
| **.MODEL SMALL**  **.STACK 100**  **.DATA**  **STRING1 DB "ENTER THE NUMBER:$"**  **STRING2 DB "PRIME$"**  **STRING3 DB "NOT PRIME$"**  **.CODE**  **MAIN PROC**  **MOV AX,@DATA**  **MOV DS,AX**  **LEA DX,STRING1**  **MOV AH,09H**  **INT 21H**  **MOV AH,01H**  **INT 21H**  **SUB AL,30H**  **CMP AL,02H**  **JE PRIME**  **CMP AL,01H**  **JE EXIT**  **CMP AL,00H**  **JE EXIT**  **MOV CL,AL**  **MOV CH,AL**  **SUB CH,01H**  **MOV BL,2**    **TOP:**  **MOV AH,00H**  **MOV AL,CL**  **DIV BL**  **CMP AH,00H**  **JE NOT\_PRIME**  **CMP BL,CH**  **JE PRIME**  **INC BL**  **JMP TOP**  **LOOP TOP** | **NOT\_PRIME:**  **MOV AH,02H**  **MOV DL,0AH**  **INT 21H**  **MOV AH,02H**  **MOV DL,0DH**  **INT 21H**  **MOV AX,@DATA**  **MOV DS,AX**  **LEA DX,STRING3**  **MOV AH,09H**  **INT 21H**  **JMP EXIT**  **LOOP NOT\_PRIME**    **PRIME:**  **MOV AH,02H**  **MOV DL,0AH**  **INT 21H**  **MOV AH,02H**  **MOV DL,0DH**  **INT 21H**  **MOV AX,@DATA**  **MOV DS,AX**  **LEA DX,STRING2**  **MOV AH,09H**  **INT 21H**  **JMP EXIT**  **LOOP PRIME**    **EXIT:**  **MOV AH,4CH**  **INT 21H**  **MAIN ENDP**  **END MAIN** |

* **Output**

Fig. Prime number checking Fig. Non-prime number checking

**Reverse order arrangement of a string**

* **Emulator Code**

|  |  |
| --- | --- |
| **.MODEL SMALL**  **.STACK 100H**  **.DATA**  **STRING1 DB 'RAJSHAHI'**  **STRING2 DB 8 DUP(?)**  **.CODE**  **MAIN PROC**  **MOV AX,00H**  **MOV AX,@DATA**  **MOV DS,AX**  **MOV ES,AX**  **LEA SI,STRING1** | **LEA DI,STRING2**  **STD**  **TOP:**  **CMP DI,00H**  **JE EXIT**  **MOVSB**  **MOV DL,[DI]**  **MOV AH,2**  **INT 21H**  **JMP TOP**  **EXIT:**  **MAIN ENDP**  **END MAIN** |

* **Output**

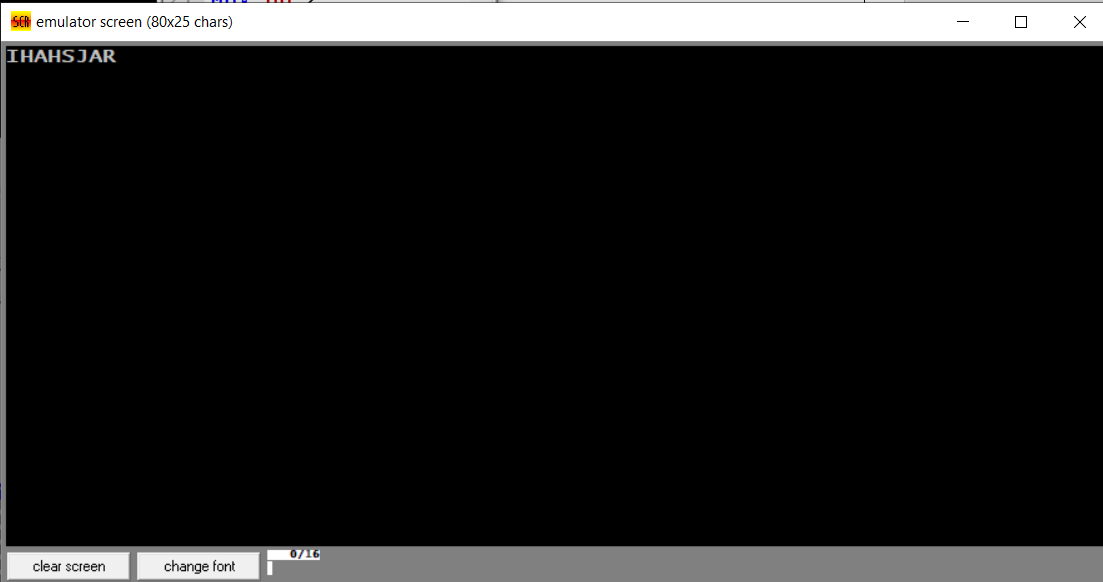
****

Fig. Reverse order arrangement of ‘RAJSHAHI’

**Result of Arithmetic operation on 7-segment display**

|  |  |
| --- | --- |
| **CODE SEGMENT**  **ASSUME CS:CODE, DS:CODE, SS:CODE,**  **ES:CODE**  **ORG 1000H; SETTING PROGRAM**  **COUNTER ON IP**  **PORTA EQU 19H**  **CONTRL EQU 1FH**  **MOV AL,10000000B**  **OUT CONTRL,AL**  **MOV AL,1**  **MOV BL,4**  **ADD AL,BL**  **CMP AL,1**  **JE L1**  **ADD AL,BL**  **CMP AL,2**  **JE L2**  **ADD AL,BL**  **CMP AL,3**  **JE L3**  **ADD AL,BL**  **CMP AL,4**  **JE L4**  **ADD AL,BL**  **CMP AL,5**  **JE L5**  **ADD AL,BL**  **CMP AL,6**  **JE L6**  **ADD AL,BL**  **CMP AL,7**  **JE L7**  **ADD AL,BL**  **MOV AL,0F8H**  **OUT PORTA,AL**  **HLT**  **L8:**  **MOV AL,80H**  **OUT PORTA,AL**  **HLT**  **L9:**  **MOV AL,90H** | **CMP AL,8**  **JE L8**  **ADD AL,BL**  **CMP AL,9**  **JE L9**  **CMP AL,0**  **JE L0**  **L1:**  **MOV AL,0F9H**  **OUT PORTA,AL**  **HLT**  **L2:**  **MOV AL,0AAH**  **OUT PORTA,AL**  **HLT**  **L3:**  **MOV AL,0B0H**  **OUT PORTA,AL**  **HLT**  **L4:**  **MOV AL,99H**  **OUT PORTA,AL**  **HLT**  **L5:**  **MOV AL,92H**  **OUT PORTA,AL**  **HLT**  **L6:**  **MOV AL,82H**  **OUT PORTA,AL**  **HLT**  **L7:**  **OUT PORTA,AL**  **HLT**  **L0:**  **MOV AL,0C0H**  **OUT PORTA,AL**  **HLT**    **CODE ENDS**  **END** |

* **Output**



Fig. Summation of ‘3’ and ‘2’ and displaying result through 7 segment

**Increasing the time delay of LED using 8255, 8254 and 8259**

|  |  |
| --- | --- |
| **INTRP PROC NEAR**  **CNT1 EQU 8001H**  **CNT2 EQU 8002H**  **CNTR EQU 8003H**  **MOV AL, 74H**  **OUT CNTR, AL**  **MOV AL, 94H**  **OUT CNTR, AL**  **MOV AL, 50H**  **OUT CNT1, AL**  **MOV AL, C3H**  **OUT CNT1, AL**  **MOV AL, 28H**  **OUT CNT2, AL**  **RET**  **INTRP ENDP**  **CODE SEGMENT**  **ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE**  **;**  **PPIC\_C EQU 1FH**  **PPIC EQU 1DH**  **PPIB EQU 1BH**  **PPIA EQU 19H**  **;**  **CTC1 EQU 0BH**  **CTCC EQU 0FH**  **;**  **INTA EQU 10H**  **INTA2 EQU INTA+2**  **OUT PPIB,AL**  **STI**  **L2: NOP**  **JMP L2**  **;**  **INT 3**  **;**  **;**  **INT\_SER:**  **SHL AH,1**  **TEST AH,00010000B**  **JNZ L1**  **OR AH,11110000B**  **JMP L3**  **; LED out**  **L1: MOV AH,11110001B**  **L3: MOV AL,AH**  **OUT PPIB,AL**  **;**  **PUSH AX**  **MOV AX,0FFFFH**  **OUT CTC1,AL**  **MOV AL,AH**  **OUT CTC1,AL**  **POP AX**  **; EOI command**  **MOV AL,00100000B**  **OUT INTA,AL**  **STI**  **IRET**  **;** | **;**  **INT\_V EQU 40H\*4**  **;**  **ORG 1000H**  **;**  **XOR BX,BX**  **MOV ES,BX**  **;**  **MOV AX,OFFSET INT\_SER**  **MOV BX,INT\_V**  **MOV WORD PTR ES:[BX],AX**  **;**  **XOR AX,AX**  **MOV WORD PTR ES:[BX+2],AX**  **;**  **CALL INIT**  **CALL P\_INIT**  **;**  **MOV AL,10000000B**  **OUT PPIC\_C,AL**  **;**  **MOV AL,11111111B**  **OUT PPIA,AL**  **;**  **MOV AL,00000000B**  **OUT PPIC,AL**  **;**  **MOV AH,11110001B**  **MOV AL,AH**  **P\_INIT PROC NEAR**  **PUSH AX**  **MOV AL,01110000B**  **OUT CTCC,AL**  **;**  **MOV AX,0FFFFH**  **OUT CTC1,AL**  **MOV AL,AH**  **OUT CTC1,AL**  **POP AX**  **RET**  **P\_INIT ENDP**  **;**  **INIT PROC NEAR**  **; ICW1**  **MOV AL,00010011B**  **OUT INTA,AL**  **;ICW2 interrupt vector**  **MOV AL,40H**  **OUT INTA2,AL**  **;ICW4**  **MOV AL,00000001B**  **OUT INTA2,AL**  **;interrupt mask**  **MOV AL,11111110B**  **OUT INTA2,AL**  **RET**  **INIT ENDP**  **;**  **CODE ENDS**  **END** |