

<p>1) Write an ALP for addition of two 8 bit unsigned numbers.</p> <pre> DATA SEGMENT NUMBER1 Db 67H NUMBER2 Db 88H SUM Db 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA       MOV DS,DX            MOV AL,NUMBER1            MOV BL,NUMBER2            ADD AL,BL            MOV SUM,AL            JNC EXIT            MOV SUM+1,01  EXIT:MOV AH, 4CH       INT 21H CODE ENDS END START </pre>	<p>2) Write an ALP for addition of two 16 bit unsigned numbers.</p> <pre> DATA SEGMENT NUMBER1 DW 6753H NUMBER2 DW 8856H SUM DW 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA       MOV DS,DX            MOV AX,NUMBER1            MOV BX,NUMBER2            SUB AX,BX            MOV SUM,AX            JNC EXIT            MOV SUM+1,01  EXIT:MOV AH, 4CH       INT 21H CODE ENDS END START </pre>
<p>3) Write an ALP for subtraction of two 8 bit unsigned numbers.</p> <pre> DATA SEGMENT NUMBER1 Db 67H NUMBER2 Db 88H SUM Db 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA       MOV DS,DX            MOV AL,NUMBER1            MOV BL,NUMBER2            SUB AL,BL            MOV SUM,AL            JNC EXIT            MOV SUM+1,01  EXIT:MOV AH, 4CH       INT 21H CODE ENDS END START </pre>	<p>4) Write an ALP for subtraction of two 16 bit unsigned numbers.</p> <pre> DATA SEGMENT NUMBER1 Db 67H NUMBER2 Db 88H SUM Db 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA       MOV DS,DX            MOV AL,NUMBER1            MOV BL,NUMBER2            SUB AL,BL            MOV SUM,AL            JNC EXIT            MOV SUM+1,01  EXIT:MOV AH, 4CH       INT 21H CODE ENDS END START </pre>
<p>5)Write an ALP for addition of two BCD numbers.</p> <pre> DATA SEGMENT NUMBER1 Db 67H NUMBER2 Db 88H </pre>	<p>6) Write an ALP for SUBTRACTION of two 16 BCD numbers.</p> <pre> DATA SEGMENT NUMBER1 DW 6753H </pre>

<pre> SUM  Db 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA         MOV DS,DX                 MOV AL,NUMBER1                 MOV BL,NUMBER2                 ADD AL,BL                 DAA                 MOV SUM,AL                 JNC EXIT                 MOV SUM+1,01  EXIT:MOV AH, 4CH INT 21H CODE ENDS END START </pre>	<pre> NUMBER2 DW 8856H SUM  DW 2 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA         MOV DS,DX                 MOV AX,NUMBER1                 MOV BX,NUMBER2                 SUB AX,BX                 DAS                 MOV SUM,AX                 JNC EXIT                 MOV SUM+1,01  EXIT:MOV AH, 4CH INT 21H CODE ENDS END START </pre>
<pre> 7) Write an ALP for division of two unsigned numbers. DATA SEGMENT NUMBER1 DW 4359H NUMBER2 DB 99H QUOTIENT DB 1 DUP(0) REMAINDER DB 1 DUP(0) DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START:MOV DX,DATA         MOV DS,DX                 MOV AX,NUMBER1                 MOV BL,NUMBER2                 DIV BL                 MOV QUOTIENT ,AL                 MOV REMAINDER,AH          MOV AH,4CH INT 21H CODE ENDS END START </pre>	
<pre> Write an ALP for multiplication of two 8-bit numbers DATA SEGMENT NUMBER1 DB 04H NUMBER2 DB 03H PRODUCT DW ? DATA ENDS  CODE SEGMENT </pre>	

<pre> ASSUME CS:CODE,DS:DATA START:MOV DX,DATA MOV DS,DX MOV AL,NUMBER1 MOV BL,NUMBER2 MUL BL MOVPRODUCT,AX MOV AH,4CH INT 21H CODE ENDS END START </pre>	
<p>8) Write an ALP for addition of array elements</p> <pre> DATA SEGMENT NUM1 DB 10H,20H,30H,40H,50H RESULT DB 1 DUP(0) CARRY DB 0H DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA MOV DX,DATA MOV DS,DX MOV CL,05H MOV SI,OFFSET NUM1 UP:MOV AL,[SI] ADD RESULT,AL JNC NEXT INC CARRY NEXT:INC SI LOOP UP  MOV AH,4CH INT 21H CODE ENDS      END </pre>	
<p>9) Write an ALP to find LARGES number from an array</p> <pre> DATA SEGMENT ARRAY DB 15H,45H,08H,56H,78H LARGEST DB 00H DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START: MOV DX,DATA MOV DS,DX MOV CX,04H MOV SI,OFFSET ARRAY MOV AL,[SI] </pre>	<p>10) Write an ALP to find LARGES number from an array</p> <pre> DATA SEGMENT ARRAY DB 15H,45H,08H,56H,78H SMALLEST DB 00H DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START: MOV DX,DATA MOV DS,DX MOV CX,04H MOV SI,OFFSET ARRAY MOV AL,[SI] </pre>

<pre> UP: INC SI     CMP AL,[SI]     JNC NEXT     MOV AL,[SI] NEXT: DEC CX     JNZ UP     MOV LARGEST,AL  MOV AH,004H INT 21H CODE ENDS    END START </pre>	<pre> UP: INC SI     CMP AL,[SI]     JC NEXT     MOV AL,[SI] NEXT: DEC CX     JNZ UP     MOV SMALLEST,AL  MOV AH,004H INT 21H CODE ENDS    END START </pre>
<p>11) Write an ALP to arrange numbers in the given array in ascending order.</p> <pre> DATA SEGMENT ARRAY DB 10H,11H,12H,13H,14H DATA ENDS  CODE SEGMENT START:ASSUME CS:CODE,DS:DATA MOV DX,DATA MOV DS,DX     MOV BL,05H     STEP1:MOV SI,OFFSET ARRAY     MOV CL,04H     STEP:MOV AL,[SI]     CMP AL,[SI+1]     JC DOWN     XCHG AL,[SI+1]     XCHG AL,[SI]     DOWN:ADD SI,1     LOOP STEP     DEC BL     JNZ STEP1  CODE ENDS END START </pre>	<p>12) Write an ALP to arrange numbers in the given array in DEscending order.</p> <pre> DATA SEGMENT ARRAY DB 10H,11H,12H,13H,14H DATA ENDS  CODE SEGMENT START:ASSUME CS:CODE,DS:DATA MOV DX,DATA MOV DS,DX     MOV BL,05H     STEP1:MOV SI,OFFSET ARRAY     MOV CL,04H     STEP:MOV AL,[SI]     CMP AL,[SI+1]     JNC DOWN     XCHG AL,[SI+1]     XCHG AL,[SI]     DOWN:ADD SI,1     LOOP STEP     DEC BL     JNZ STEP1  CODE ENDS END START </pre>
<p>13) Write an ALP to transfer block of data using string instruction.</p> <pre> DATA SEGMENT ARR1 DB 05H,06H,07H,09H,02H DATA ENDS  EXTRA SEGMENT ARR2 DB 10 DUP(0) EXTRA ENDS  CODE SEGMENT ASSUME DS:DATA,CS:CODE,ES:EXTRA START:MOV AX,DATA     MOV DS,AX     MOV AX,EXTRA     MOV EX,AX </pre>	<p>14) Write an ALP to transfer block of data without using string instruction.</p> <pre> DATA SEGMENT ARR1 DB 05H,06H,07H,09H,02H DATA ENDS  EXTRA SEGMENT ARR2 DB 10 DUP(0) EXTRA ENDS  CODE SEGMENT ASSUME DS:DATA,CS:CODE,ES:EXTRA START:MOV AX,DATA     MOV DS,AX     MOV AX,EXTRA     MOV EX,AX </pre>

<pre> MOV CX,03 LEA SI,ARR1 LEA DI,ARR2 CLD REP MOVSB  MOV AH,4CH INT 21H CODE ENDS END START </pre>	<pre> MOV CL,05 LEA SI,ARR1 LEA DI,ARR2 UP: MOV AL,[SI] MOV [DI],AL INC SI INC DI LOOP UP  MOV AH,4CH INT 21H CODE ENDS END START </pre>
<p>15) Write an ALP to compare two string using string instruction.</p> <pre> Data Segment str1 db 'GLSICT','\$' strlen1 db \$-str1 str2 db 'GLSICT','\$' strlen2 db \$-str2 streq db 'Strings are Equal','\$' struneq db 'Strings are Unequal','\$' Data Ends CODE SEGMENT ASSUME CS:CODE, DS:DATA BEGIN: MOV AX, DATA MOV DS, AX MOV ES, AX  LEA SI, STR1 LEA DI, STR2 MOV CX, 6 MOV AL, STRLEN1 MOV BL, STRLEN2 CMP AL, BL JNE NOT_EQUAL  REPE CMPSB JNE NOT_EQUAL JMP EQUAL  NOT_EQUAL: MOV AH, 09H LEA DX, STRUNEQ INT 21H JMP EXIT EQUAL: MOV AH, 09H LEA DX, STREQ </pre>	<p>16) WRITE AN ALP TO DISLAY STRING IN REVERSE ORDER</p> <pre> DATA SEGMENT STRING DB "STUPID\$" ARR DB 10 DUP('\$') DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START: MOV DX,DATA MOV DS,DX MOV CL,00H LEA SI,STRING MOV BL,'\$'  AGAIN: MOV AL,[SI] CMP AL,BL JZ REVERSE INC CL INC SI JMP AGAIN  REVERSE: LEA DI,ARR DEC SI UP: MOV AL,[SI] MOV [DI],AL INC DI DEC SI LOOP UP  MOV [DI],'\$'  LEA DX,ARR MOV AH,09H INT 21H </pre>

<pre> INT 21H EXIT: MOV AH, 4CH INT 21H CODE ENDS  END BEGIN </pre>	<pre> MOV AH,4CH INT 21H CODE ENDS END START </pre>
<pre> <b>17) WRITE AN ALP TO FIND(COUNT) STRING LENGTH .</b> DATA SEGMENT STRING DB "INFO\$" LEN DB ? DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START: MOV DX,DATA MOV DS,DX  MOV CL,00H LEA SI,STRING MOV BL,'\$'  AGAIN: MOV AL,[SI] CMP AL,BL JZ EXIT INC CL INC SI JMP AGAIN EXIT:  MOV LEN,AL MOV AH,4CH  INT 21H CODE ENDS END START </pre>	<pre> <b>18) WRITE AN ALP TO CONCATENATE TWO STRINGS.</b> DATA SEGMENT STR1 DB "COMPUTER\$" STR2 DB 'TECHNOLOGY\$' STR3 DB 25 DUP('\$') DATA ENDS  CODE SEGMENT ASSUME CS:CODE,DS:DATA START: MOV DX,DATA MOV DS,DX  LEA SI,STR1 LEA DI,STR3 MOV BL,'\$' AGAIN: MOV AL,[SI] CMP AL,BL JZ GO MOV [DI],AL INC SI INC DI JMP AGAIN GO:  LEA SI,STR2 AGAIN1: MOV AL,[SI] CMP AL,BL JZ EXIT MOV [DI],AL INC SI INC DI JMP AGAIN1 EXIT: MOV [DI],'\$' LEA DX,STR3 MOV AH,09H INT 21H  MOV AH,4CH INT 21H CODE ENDS  END START </pre>