1. **AIM**: Write an ALP to transfer data.
   string source starting address 0X40000000 to string destination starting address 0X40000010, starting length 9 Bytes.

   Operatus: PC and Keil µVision 4.0 software

   Procedure :

   Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

   Step 2: Give a proper name to your project and save it with extension **.asm**

   Step 3: Now select a ARM based processor **LPC2148** and click on ok.

   Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

   Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

   Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

   Step 7: Click on debug start debug section, for checking your code is running or not.

   Step 8: Enter input values into memory locations, Run your program and verify output.

   PROGRAM

```
AREA DATA_TRANS,CODE,READONLY
        ENTRY
        LDR R0,=0X40000000 ; String source starting address
        LDR R1,=0X40000010 ; String Destination  starting address
         MOV R4,#09            :  String length
LOOP  LDRB R2,[R0],#01      ; Load byte data into R2 register from memory
```

```
        STRB R2,[R1],#01     ; Store byte data from R2 register to memory
        SUBS R4,#01          ; Decrement string length by one
        CMP R4,#00           ; Compare string counter with zero
        BNE LOOP
STOP    B       STOP
        END
```

Input: memory location starting address 0X40000000 : 23,54,67,87,36,89,92,48,68

Output: memory location starting address 0X40000010 : 23,54,67,87,36,89,92,48,68

RESULT: verified data transfer from one area to another area in memory.

2. **AIM**: Write an ALP to Exchange data

   String source starting address 0X40000000 and string destination starting address 0X40000010, starting length 9 Bytes.

   Operatus: PC and Keil µVision 4.0 software

   Procedure :

   Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

   Step 2: Give a proper name to your project and save it with extension **.asm**

   Step 3: Now select a ARM based processor **LPC2148** and click on ok.

   Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

   Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA DATA_EXCH,CODE,READONLY
        ENTRY
        LDR R0,=0X40000000
        LDR R1,=0X40000010
        MOV R4,#09
LOOP  LDRB R5,[R0]
        LDRB R6,[R1]
        STRB R6,[R0],#01
        STRB R5,[R1],#01
         SUBS R4,#01
        CMP R4,#00
        BNE LOOP
STOP  B       STOP
    END
```

Inputs : memory location starting address 0X40000000 : 23,54,67,87,36,89,92,48,68

memory location starting address 0X40000010 : 1,2,3,4,5,6,7,8,9

Output: memory location starting address 0X40000000: 1,2,3,4,5,6,7,8,9

memory location starting address 0X40000010 : 23,54,67,87,36,89,92,48,68

RESULT: verified data exchange from one area to another area in memory.

3. **AIM**: Write an ALP to find sum of N numbers. String starting address 0X40000000 and starting length 5 words.

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA NADD, CODE, READONLY
        ENTRY
        MOV R1,#00              ; upper 32 bit result counter
        LDR R0,=0X40000000      ; string starting address
        MOV R2,#04              ;string length
        LDR R3,[R0],#04         ; Load string first word into R3 register
```

```
BACK      LDR R4,[R0],#04     ; Load string next word into R4 register
          ADDS R3,R3,R4       ; add next word with previous word
          BHI COUNTER         ; monitor carry bit
          B NEXT
COUNTER   ADD R1,R1,#01
NEXT       SUBS R2,R2,#01     ; Decrement  string length size by one
          CMP R2,#00          ; compare string length with zoer
          BNE BACK
           STR R1,[R0],#04    ; store higher 32 bit result in memory
           STR R3,[R0]        ; store lower 32 bit result in memory
STOP       B       STOP
          END
```

Inputs : memory location starting address 0X40000000 : 0xff, 0xff

Output: memory location starting address 0X40000014: 00 00 00 04

memory location starting address 0X40000015 : FF FF FF FB

RESULT: verified data exchange from one area to another area in memory.

4. **AIM**:. Write an ALP to find smallest number from the given string, String starting address 0X40000000, string length 7 bytes, store smallest number into 0X40000010 location.

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM,CODE,READONLY
        ENTRY
        LDR R0,=0X40000000          ; string starting address
        MOV R1,#6                   ; string length counter
        LDR R2,=0X40000010          ; smallest number  storage location
        LDRB R3,[R0],#01            ;Load first data in R3 register
```

```
UP      LDRB R4,[R0],#01        ; Load next data in R3 register
        CMP R3,R4               ;compare data
        MOVHI R3,R4             ; move data from R4 to R3 if R3 is greater than R4
        SUB R1,R1,#01           ; decrement string counter by one
        CMP R1,#0
        BNE UP
        STRB R3,[R2]
STOP        B       STOP
        END
```

Inputs : memory location starting address 0X40000000 : 67,45,45,87,13,98,23

Output: memory location starting address 0X40000010: 13

RESULT: verified data exchange from one area to another area in memory.

5. **AIM**:. Write an ALP to find largest number from the given string, String starting address 0X40000000, string length 7 bytes, store smallest number into 0X40000010 location

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM,CODE,READONLY
        ENTRY
        LDR R0,=0X40000000      ; String starting address.
        MOV R1,#6               ;String length counter.
        LDR R2,=0X40000010      ; Store largest number in this address.
        LDRB R3,[R0],#01        ;Load  first data into R3 register from memory.
UP      LDRB R4,[R0],#01        ; Load next byte from next memory location.
        CMP R3,R4               ;compare two data values

        MOVLS   R3,R4           ; move data from R4 to R3 when R3 is lessthan or

                                  same

        SUB R1,R1,#01           ; decrement string length counter by one
        CMP R1,#0
        BNE UP
        STRB R3,[R2]
STOP        B       STOP
        END
```

Inputs : memory location starting address 0X40000000 : 67,45,45,87,13,98,23

Output: memory location starting address 0X40000010: 87

RESULT: verified data exchange from one area to another area in memory.

6. **AIM**:. Write an ALP to arrange ascending order from the given string, String starting address 0X40000000, string length 7 bytes.

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM, CODE, READONLY
                ENTRY
                MOV R5,#0X6              ; outer counter (N-1) iterations
AGNN            LDR R0,=0X40000000       ;string starting address
                MOV R6,#0X6              ; inner counter (N-1) iterations
AGN             LDRB R1,[R0],#01         ; load first byte from memory to R1 register
                LDRB R2,[R0]             ; load next byte from memory to R2 register
                CMP R1,R2                ; compare two data values
                BLS NEXT                 ;if R1 is less than or same R2 jump to target
                STRB R1,[R0]             ; exchange data from R1 to memory
                STRB R2,[R0,#-01]        ; exchange data from R2 to memory
NEXT            SUB R6,R6,#01            ;decrement inner counter by one
                CMP R6,#00
                BNE  AGN
                SUB R5,R5,#01            ; decrement outer counter by one
                CMP R5,#00
                BNE  AGNN
L               B        L
                END
```

Inputs : memory location starting address 0X40000000 : 67,45,45,87,13,98,23

Output: memory location starting address 0X40000000: 13,23,45,45,67,87,98

RESULT: verified data exchange from one area to another area in memory.

7. **AIM**:. Write an ALP to arrange Descending order from the given string, String starting address 0X40000000, string length 7 bytes.

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM, CODE, READONLY
                ENTRY
                MOV R5,#0X6             ; outer counter (N-1) iterations
AGNN            LDR R0,=0X40000000      ; string starting address
                MOV R6,#0X6             ; inner counter (N-1) iterations
AGN             LDRB R1,[R0],#01        ; load first byte from memory to R1 register
                LDRB R2,[R0]            ; load next byte from memory to R2 register
                CMP R1,R2               ; compare two data values
                BHI   NEXT              ;if R1 is greater than R2 jump to target
                STRB R1,[R0]            ; exchange data from R1 to memory
                STRB R2,[R0,#-01]       ; exchange data from R2 to memory
NEXT            SUB R6,R6,#01           ; decrement inner counter by one
                CMP R6,#00
                BNE  AGN
```

```
        SUB R5,R5,#01          ;decrement outer counter by one
        CMP R5,#00
        BNE  AGNN
L       B       L
        END
```

Inputs : memory location starting address 0X40000000 : 67,45,45,87,13,98,23

Output: memory location starting address 0X40000000: 98,87,67,45,45,23,13

RESULT: verified data exchange from one area to another area in memory.

8. **AIM**:. Write an ALP to separate even and odd numbers from the given string, also count even and odd numbers.
   String length 8 bytes
   source String starting address 0X40000000
   even numbers starting address 0X40000010
   odd numbers starting address 0X40000020

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM, CODE, READONLY
                ENTRY
                MOV R2,#0           ;Odd numbers counter
                MOV R3,#0           ;Even numbers counter
                MOV R1,#8           ; String length counter
                LDR R0,=0X40000000      ; String  source starting address
                LDR R7,=0X40000010      ; Store odd numbers from this address
                LDR R8,=0X40000020      ; Store even numbers from this address
LOOP2           LDRB R5,[R0],#1         ; Load data from source address
                MOV R6,R5
                MOVS  R5,R5,ROR #1      ; Verify data even or odd
                BHI ODD                 ; if data is odd number jump to target
                ADDS R3,R3,#1           ;increment even numbers counter
                STRB R6,[R8],#1
                B LOOP1
ODD            ADD R2,R2,#1             ;increment odd numbers counter
                STRB R6,[R7],#1
LOOP1           SUB R1,R1,#1            ; Verify string length counter
                CMP R1,#00
                BNE    LOOP2
L               B       L
                END
```

Inputs : memory location starting address 0X40000000 : 1,2,3,4,5,6,7,8

Output: memory location odd numbers starting address 0X40000010: 1,3,5,7

memory location even numbers starting address 0X40000020: 2,4,6,8

RESULT: verified data exchange from one area to another area in memory.

9. **AIM**:. Write an ALP to find No of 1's and 0's in given Number
   Data address 0X40000000

Operatus: PC and Keil µVision 4.0 software

Procedure :

Step 1: Open your Keil Microvision4 software, click on project select new Microvision project.

Step 2: Give a proper name to your project and save it with extension **.asm**

Step 3: Now select a ARM based processor **LPC2148** and click on ok.

Step 4: A project window will get open on the right side you will see target 1 click on that and then double click on source group 1, then select your previously saved file and add it to the source group and close it.

Step 5: Now double click on the file that is visible below the source group 1, a plain text area will get open.

Step 6: Type your program in the plain text area and then click on build for checking errors then click on rebuild.

Step 7: Click on debug start debug section, for checking your code is running or not.

Step 8: Enter input values into memory locations, Run your program and verify output.

PROGRAM

```
AREA GITAM, CODE, READONLY
```

```
        ENTRY
    MOV R2,#0                   ; 1's counter
    MOV R3,#0                   ;0's counter
    MOV R1,#32                  ; Data bits counter
    LDR R0,=0X40000000          ; Data available address
    LDR R5,[R0]                 ; Load data from memory to R5 register
LOOP2   MOVS R5,R5,ROR #1 ; verify bit zero or one
        BHI ONES                ; if bit is one jump to target address
        ADD   R3,R3,#1          ; increment  zoer's counter
        B     LOOP1
ONES    ADD R2,R2,#1            ; increment one's counter
LOOP1   SUB R1,R1,#1
        CMP R1,#00
        BNE LOOP2
L       B       L
        END
```

Inputs : memory location starting address 0X40000000 : 23,34,45,66

Output: R2=11, R3=21

RESULT: verified data exchange from one area to another area in memory.