**Ayush Goyal**

**190905522 CSE D 62**

**ES Lab-5 (Week-5) – Sorting and Searching Programs**

1. **Write an assembly program to sort an array using selection sort.**

**CODE:**

AREA RESET,DATA,READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000

DCD Reset\_Handler

ALIGN

AREA mycode,CODE,READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =SRC

LDR R1, =S

LDR R2,[R1]

LDR R7, =DST

MOV R8,#0

Up CMP R8,R2

BEQ Wod

ADD R8,#1

LDR R9,[R0],#4

STR R9,[R7],#4

B Up

Wod LDR R0,=DST

MOV R1, R0

MOV R3,R0

MOV R10,#0

MOV R11,#0

Com CMP R11, R2

BEQ STOP

ADD R3,R0,#4

MOV R1,R0

ADD R10,R11,#1

Moc CMP R10,R2

BEQ Dow

ADD R10,#1

LDR R4,[R3],#4

LDR R5,[R1]

CMP R5,R4

BLT Moc

MOV R1,R3

SUB R1,#4

B Moc

Dow ADD R11,#1

LDR R4,[R0]

LDR R5,[R1]

STR R4,[R1]

STR R5,[R0],#4

B Com

STOP B STOP

S DCD 0xA

SRC DCD 0x30,0x29,0x28,0x27,0x26,0x25,0x24,0x23,0x22,0x21

AREA mydata,DATA,READWRITE

DST DCD 0,0,0,0,0,0,0,0,0,0

END

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

1. **Write an assembly program to find the factorial of an unsigned number using recursion.**

**CODE:**

AREA RESET, CODE, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000

DCD Reset\_Handler

AREA myCode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =SRC

LDR R0, [R0]

LDR R1, =1

LDR R2, =0

BL Rec

LDR R0, =DST

STR R1, [R0]

STOP B STOP

Rec PUSH {R2, LR}

ADD R2, #1

CMP R2, R0

BHI Sto

BL Rec

MUL R1, R2

Sto POP {R2, LR}

BX LR

SRC DCD 5

AREA MYDATA, DATA, READWRITE

DST DCD 0

END

**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

1. **Write an assembly program to search an element in an array of ten 32-bit numbers using linear search.**

**CODE:**

AREA RESET, CODE, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000

DCD Reset\_Handler

AREA myCode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

LDR R0, =SRC

LDR R2, =KEY

MOV R4, #9

LDR R3, [R2]

Up TEQ R4, #0

BEQ Don

LDR R1, [R0], #4

TEQ R1, R3

BEQ Fon

SUB R4, #1

B Up

Fon MOV R5, #1

Don TEQ R5, #1

BEQ Yes

MOV R4, #0

B No

Yes RSB R4, #10

LDR R6, =DST

SUB R4,#1

STRB R5, [R6], #1 ;Stores 1 for found, 0 for not found

STRB R4, [R6] ;Stores Index Value of the number if found

B STOP

No LDR R6, =DST

STR R7, [R6]

STOP B STOP

SRC DCD 34, 25, 23, 46, 15, 46, 27, 98, 89

KEY DCD 23

AREA MYDATA, DATA, READWRITE

DST DCD 0

END

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

1. **Assume that ten 32-bit numbers are stored in registers R1-R10. Sort these numbers in the empty ascending stack using selection sort and store the sorted array back into the registers. Use STM and LDMDB instructions wherever necessary.**

**CODE:**

AREA RESET, CODE, READONLY

EXPORT \_\_Vectors

\_\_Vectors

DCD 0x10001000

DCD Reset\_Handler

AREA myCode, CODE, READONLY

ENTRY

EXPORT Reset\_Handler

Reset\_Handler

MOV R1, #6

MOV R2, #10

MOV R3, #7

MOV R4, #1

MOV R5, #5

MOV R6, #8

MOV R7, #9

MOV R8, #3

MOV R9, #2

MOV R10, #4

STMEA R13!, {R1-R10}

BL Srt ; Sort Call

LDMEA R13!, {R1-R10}

STOP B STOP

Srt MOV R2, #0 ; I

MOV R3, #0 ; J

Top CMP R2, #36

BEQ Sto

SUB R8, R13, R2

SUB R8, #4

LDR R4, [R8]

MOV R7, R2

MOV R3, R2

ADD R3, #4

UP CMP R3, #40

BEQ Hel

SUB R8, R13, R3

SUB R8, #4

LDR R5, [R8]

CMP R4, R5

BCS Leh

SUB R8, R13, R3

SUB R8, #4

LDR R4, [R8]

MOV R7, R3

Leh ADD R3, #4

B UP

Hel SUB R8, R13, R7

SUB R8, #4

LDR R4, [R8]

SUB R8, R13, R2

SUB R8, #4

LDR R5, [R8]

SUB R8, R13, R7

SUB R8, #4

STR R5, [R8]

SUB R8, R13, R2

SUB R8, #4

STR R4, [R8]

ADD R2, #4

B Top

Sto BX LR

END

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**THE END**