5

PES UNIVERSITY

Department of Computer Science & Engineering

Microprocessor & Computer Architecture Lab

UE23CS251B

WEEK 5 submission

Name of the Student	*CHIRAG K M*
SRN	*PES1UG23CS167*
Section	*C*
Department	*CSE*
Campus	*RR*

Q1. Write an ALP using ARM7TDMI to multiply two matrix.

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

matrixmul.s

```
                .DATA
00001088:               A: .WORD 1,2,3,4,5,6,7,8,9
000010AC:               B: .WORD 1,2,3,4,5,6,7,8,9
000010D0:               C: .WORD 0,0,0,0,0,0,0,0,0

                .TEXT
00001000:E59F0074       LDR R0,=A
00001004:E59F1074       LDR R1,=B
00001008:E59F2074       LDR R2,=C

0000100C:E3A03000       MOV R3,#0  ;INNER LOOP COUNT I INDEX
00001010:E3A04000       MOV R4,#0  ;OUTER LOOP COUNT J INDEX
00001014:E3A0A003       MOV R10,#3 ; NUMBER OF ELEMENTS IN A ROW
00001018:E3A08000       MOV R8,#0 ;VALUE OF K

0000101C:E02B8A93   LOOP1:MLA R11,R3,R10,R8
00001020:E1A0B10B       MOV R11,R11,LSL #2
00001024:E790500B           LDR R5,[R0,R11]
00001028:E02C4A98           MLA R12,R8,R10,R4
0000102C:E1A0C10C       MOV R12,R12,LSL #2
00001030:E791600C           LDR R6,[R1,R12]

00001034:E00B0695           MUL R11,R5,R6 ; REGISTER R11 IS REUSED.
00001038:E089900B           ADD R9,R9,R11

0000103C:E2888001           ADD R8,R8,#1    ; INCREMENT K  INNERMOST LOOP
00001040:E3580003           CMP R8,#3
00001044:1AFFFFF4           BNE LOOP1

00001048:E02C4A93           MLA R12,R3,R10,R4     ; STORE THE IN C[I][J]
0000104C:E1A0C10C       MOV R12,R12,LSL #2
00001050:E782900C           STR R9,[R2,R12]

00001054:E3A08000           MOV R8,#0       ; K=0
00001058:E3A09000           MOV R9,#0       ; C[I][J]=0
0000105C:E2844001           ADD R4,R4,#1
00001060:E3540003           CMP R4,#3
00001064:1AFFFFEC       BNE LOOP1
00001068:E3A04000           MOV R4,#0
0000106C:E2833001           ADD R3,R3,#1
00001070:E3530003           CMP R3,#3
00001074:1AFFFFE8           BNE LOOP1
00001078:EF000011           SWI 0X011
                .END
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0      :4232
R1      :4268
R2      :4304
R3      :3
R4      :0
R5      :9
R6      :9
R7      :0
R8      :0
R9      :0
R10(sl):3
R11(fp):81
R12(ip):32
R13(sp):21504
R14(lr):0
R15(pc):4216
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

MemoryView0

Word Size: 8Bit | 16Bit | 32Bit

00001088

```
00001088  00000001  00000002  00000003
00001094  00000004  00000005  00000006
000010A0  00000007  00000008  00000009
```

MemoryView0

Word Size: 8Bit | 16Bit | 32Bit

000010AC

```
000010AC  00000001  00000002  00000003
000010B8  00000004  00000005  00000006
000010C4  00000007  00000008  00000009
```

MemoryView0

Word Size: 8Bit | 16Bit | 32Bit

000010D0

```
000010D0  0000001E  00000024  0000002A
000010DC  00000042  00000051  00000060
000010E8  00000066  0000007E  00000096
```

Q2. Write an ALP using ARM7TDMI to find the length of string

stringcnt.s

```
                        .text
00001000:E59F0014               LDR R0,=A
00001004:E3A05000               MOV R5,#0

00001008:                       LOOP:
00001008:E4D01001                       LDRB R1,[R0],#1
0000100C:E3510000                       CMP R1,#0
00001010:12855001                       ADDNE R5,R5,#1
00001014:1AFFFFFB                       BNE LOOP
00001018:EF000011               SWI 0X011
                        .data
00001020:                       A: .ASCIZ "ABCDEFGHIJJ"
```

RegistersView

General Purpose | Floating Point

Hexadecimal | Unsigned Decimal | Signed Decimal

```
R0      :4140
R1      :0
R2      :0
R3      :0
R4      :0
R5      :11
R6      :0
R7      :0
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4120
------------------
CPSR Register
Negative(N):0
Zero(Z)    :1
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x600000df
```

MemoryView1

00001020

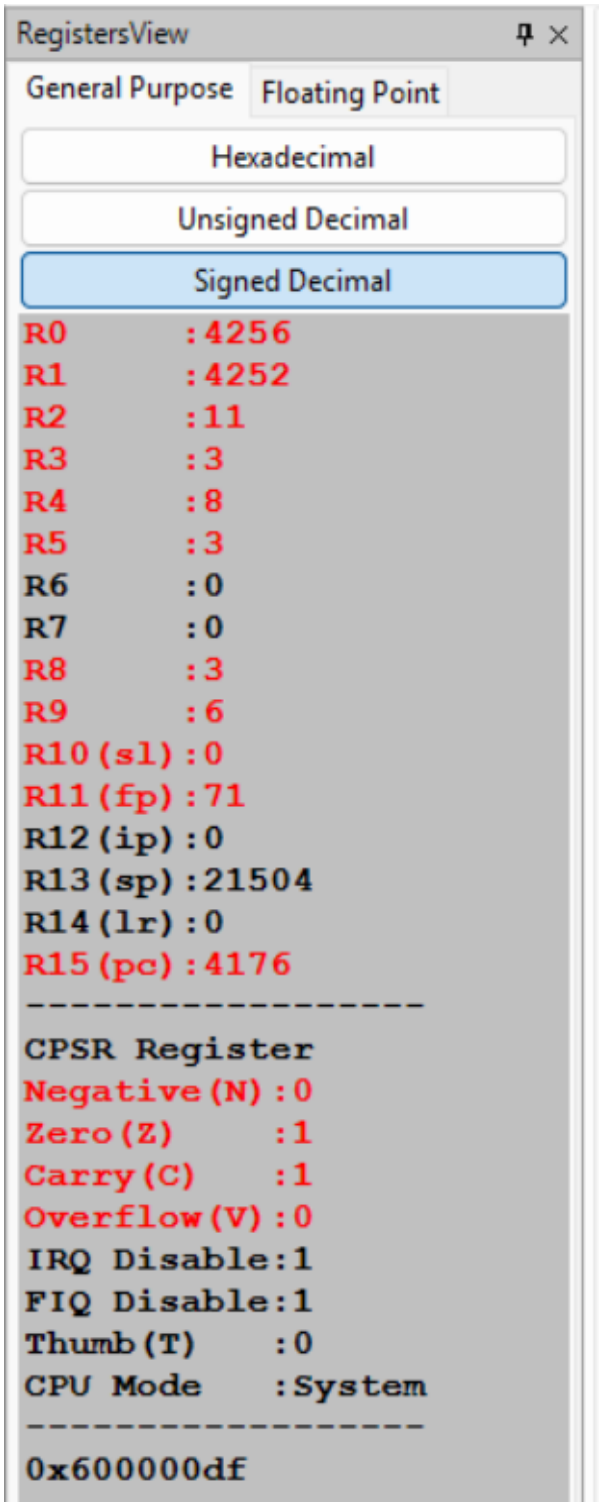
00001020	44434241	48474645	004A4A49	81818181
00001054	81818181	81818181	81818181	81818181
00001088	81818181	81818181	81818181	81818181
000010BC	81818181	81818181	81818181	81818181

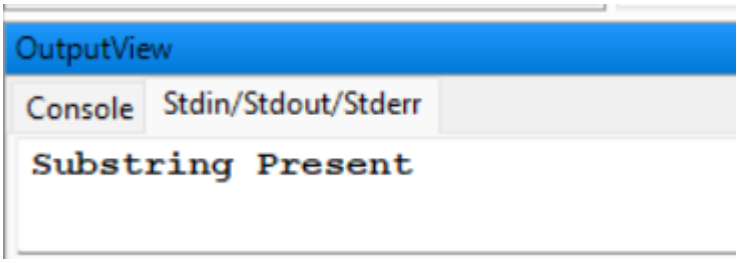
Q3. Write an ALP using ARM7TDMI to find the substring present or not.

substr.s

```
                    .text
00001000:E59F0078           LDR R0,=A
00001004:E59F1078           LDR R1,=B

00001008:E3A0200B           MOV R2,#11              ; text length(n)
0000100C:E3A03003           MOV R3,#3               ; pattern length(m)
00001010:E0424003           SUB R4,R2,R3            ; R4 = n-m

00001014:E3A05000           MOV R5,#0               ; j
00001018:E3A08000           MOV R8,#0               ; i

0000101C:                   LOOP:
0000101C:E3540000                   CMP R4,#0

00001020:0A00000B                   BEQ FAIL

00001024:E3A05000                   MOV R5,#0
00001028:E1550003                   CMP R5,R3
0000102C:BA00000C                   BLT WLOOP
00001030:                   CONTINUE:
00001030:E1550003                   CMP R5,R3
00001034:0A000003                   BEQ EXIT

00001038:E1580004                   CMP R8,R4
0000103C:B2888001                   ADDLT R8,R8,#1
00001040:BAFFFFF5                   BLT LOOP
00001044:EA000002                   B FAIL


00001048:                   EXIT:
00001048:E59F0038                   LDR R0,=C
0000104C:EF000002                   SWI 0X02
00001050:EF000011                   SWI 0X011
00001054:                   FAIL:
00001054:E3E08000                   MOV R8,#-1
00001058:E59F002C                   LDR R0,=D
0000105C:EF000002                   SWI 0X02
00001060:EAFFFFF8                   B EXIT
```

```
00001060:EAFFFFF8                   B EXIT
00001064:                   WLOOP:
00001064:E7D1A005                   LDRB R10,[R1,R5]
00001068:E0859008                   ADD R9,R5,R8
0000106C:E7D0B009                   LDRB R11,[R0,R9]
00001070:E15B000A                   CMP R11,R10
00001074:02855001                   ADDEQ R5,R5,#1
00001078:0AFFFFF9                   BEQ WLOOP
0000107C:EAFFFFEB                   B CONTINUE



                    .data
00001090:                   A: .ASCIZ "ABCDEFGHIJJ"
0000109C:                   B: .ASCIZ "DEF"
000010A0:                   C: .ASCIZ "Substring Present"
000010B2:                   D: .ASCIZ "Not Present"
```

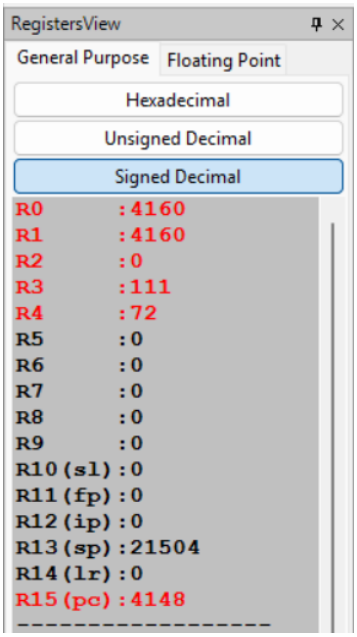

RegistersView
General Purpose
Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :4256
R1 :4252
R2 :11
R3 :3
R4 :8
R5 :3
R6 :0
R7 :0
R8 :3
R9 :6
R10(sl):0
R11(fp):71
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4176
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df

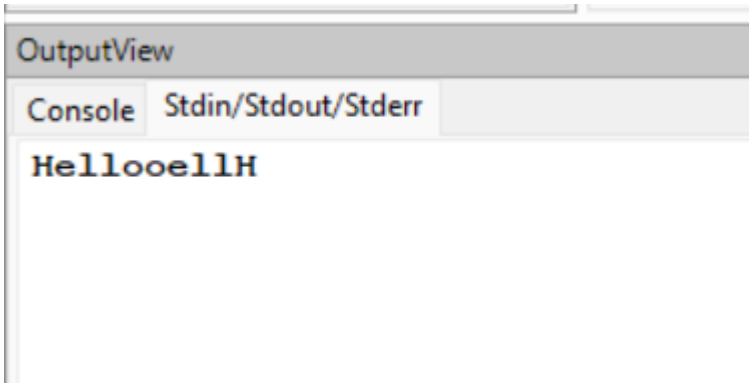

OutputView
Console
Stdin/Stdout/Stderr
Substring Present

Q4. Write a program to swap the first and last character of a given string.

swap.s

```
                        .text
00001000:E3A00D41           LDR R0, =A
00001004:EF000002           SWI 0x02

00001008:E1A01000           MOV R1, R0

0000100C:               FIND_END:
0000100C:E4D02001           LDRB R2, [R0], #1
00001010:E3520000           CMP R2, #0
00001014:1AFFFFFC           BNE FIND_END

00001018:E2400002           SUB R0, R0, #2
0000101C:E5D03000           LDRB R3, [R0]
00001020:E5D14000           LDRB R4, [R1]
00001024:E5C13000           STRB R3, [R1]
00001028:E5C04000           STRB R4, [R0]

0000102C:E3A00D41           LDR R0,=A
00001030:EF000002           SWI 0x02
00001034:EF000011           SWI 0x011

                        .data
00001040:               A: .ASCIZ "Hello"
```





THANK YOU