Microprocessor and Computer Architecture (MPCA) Laboratory
UE20CS252 4th Semester,
Academic Year 2021-22

Kumar Abhimanyu
PES1UG20CS224
Date: 28/01/2022

Week # 1

Title of the Program

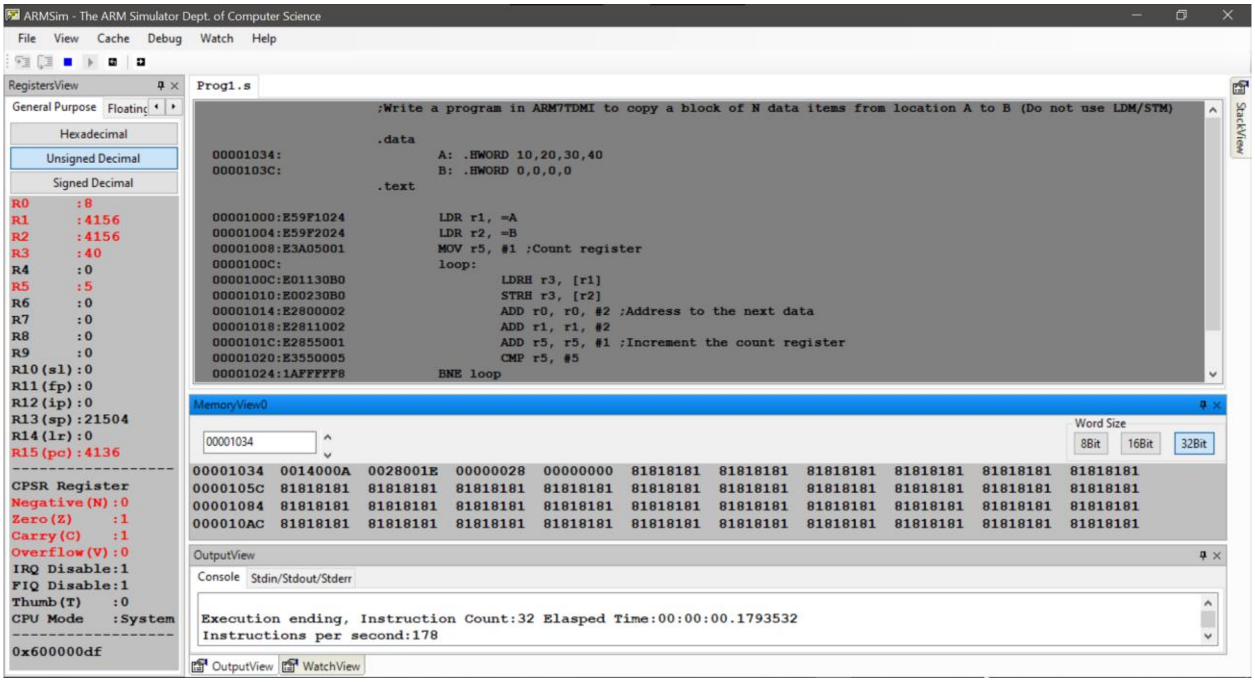
Write a program in ARM7TDMI to copy a block of N data items from location A to B with HalfWord (HWORD) (Do not use LDM/STM)

Program Code

```
.data
        A: .HWORD 10,20,30,40
        B: .HWORD 0,0,0,0
.text

        LDR r1, =A
        LDR r2, =B
        MOV r5, #1 ;Count register
        loop:
                LDRH r3, [r1]
                STRH r3, [r2]
                ADD r0, r0, #2 ;Address to the next data
                ADD r1, r1, #2
                ADD r5, r5, #1 ;Increment the count register
                CMP r5, #5
                BNE loop
                SWI 0X011
.end
```

Screenshot of ArmSimulator of the Program Executed



```
                         ;Write a program in ARM7TDMI to copy a block of N data items from location A to B (Do not use LDM/STM)

                         .data
00001034:                        A: .HWORD 10,20,30,40
0000103C:                        B: .HWORD 0,0,0,0
                         .text

00001000:E59F1024                LDR r1, =A
00001004:E59F2024                LDR r2, =B
00001008:E3A05001                MOV r5, #1 ;Count register
0000100C:                        loop:
0000100C:E01130B0                        LDRH r3, [r1]
00001010:E00230B0                        STRH r3, [r2]
00001014:E2800002                        ADD r0, r0, #2 ;Address to the next data
00001018:E2811002                        ADD r1, r1, #2
0000101C:E2855001                        ADD r5, r5, #1 ;Increment the count register
00001020:E3550005                        CMP r5, #5
00001024:1AFFFFF8                BNE loop
```

```
R0       :8
R1       :4156
R2       :4156
R3       :40
R4       :0
R5       :5
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) :4136
-------------------
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
```

```
00001034
00001034  0014000A  0028001E  00000028  00000000  81818181  81818181  81818181  81818181  81818181  81818181
0000105C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001084  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:32 Elasped Time:00:00:00.1793532
Instructions per second:178
```

Title of the Program

Program to transfer a block of data from location A to B using BYTE

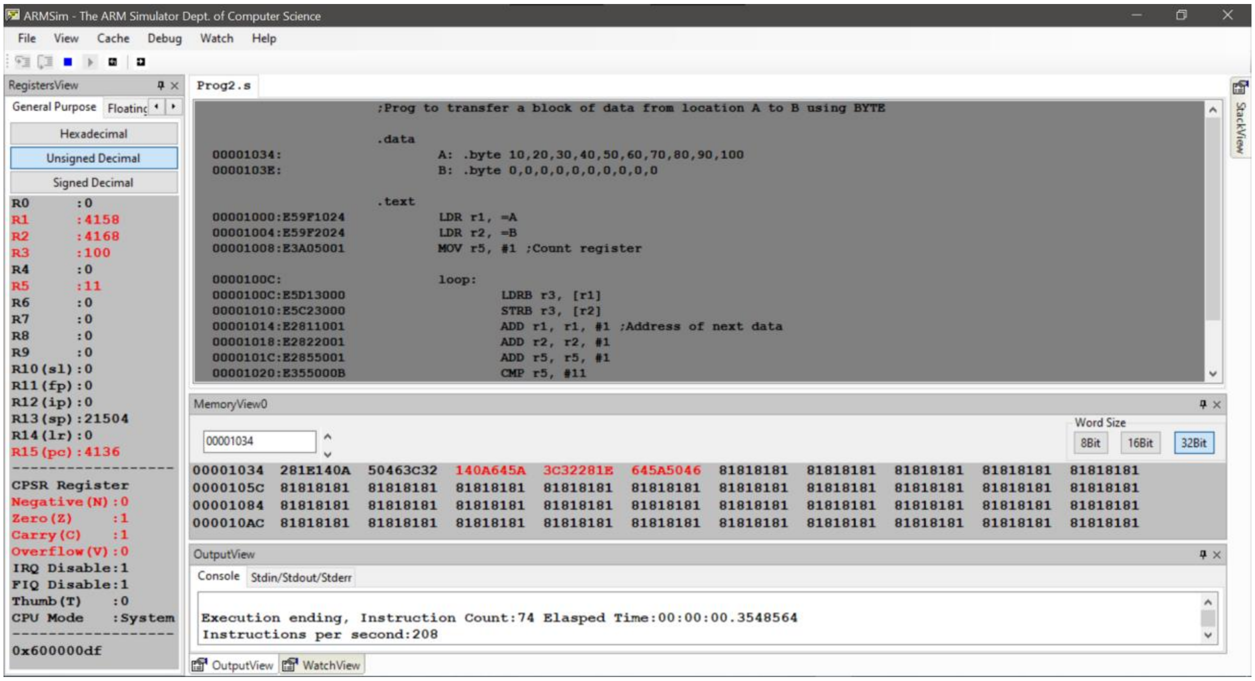
Program Code

```
.data
        A: .byte 10,20,30,40,50,60,70,80,90,100
        B: .byte 0,0,0,0,0,0,0,0,0,0
```

```
.text
        LDR r1, =A
        LDR r2, =B
        MOV r5, #1 ;Count register

        loop:
                LDRB r3, [r1]
                STRB r3, [r2]
                ADD r1, r1, #1 ;Address of next data
                ADD r2, r2, #1
                ADD r5, r5, #1
                CMP r5, #11
                BNE loop
        SWI 0X011 ; Logical end of the program
.end
```

Screenshot of ArmSimulator of the Program Executed



```
R0        :0
R1        :4158
R2        :4168
R3        :100
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):4136
-------------------
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
```

```
                        ;Prog to transfer a block of data from location A to B using BYTE

                        .data
00001034:                       A: .byte 10,20,30,40,50,60,70,80,90,100
0000103E:                       B: .byte 0,0,0,0,0,0,0,0,0,0

                        .text
00001000:E59F1024               LDR r1, =A
00001004:E59F2024               LDR r2, =B
00001008:E3A05001               MOV r5, #1 ;Count register

0000100C:                       loop:
0000100C:E5D13000                       LDRB r3, [r1]
00001010:E5C23000                       STRB r3, [r2]
00001014:E2811001                       ADD r1, r1, #1 ;Address of next data
00001018:E2822001                       ADD r2, r2, #1
0000101C:E2855001                       ADD r5, r5, #1
00001020:E355000B                       CMP r5, #11
```

```
00001034
00001034  281E140A  50463C32  140A645A  3C32281E  645A5046  81818181  81818181  81818181  81818181  81818181
0000105C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001084  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:74 Elasped Time:00:00:00.3548564
Instructions per second:208
```

Title of the Program

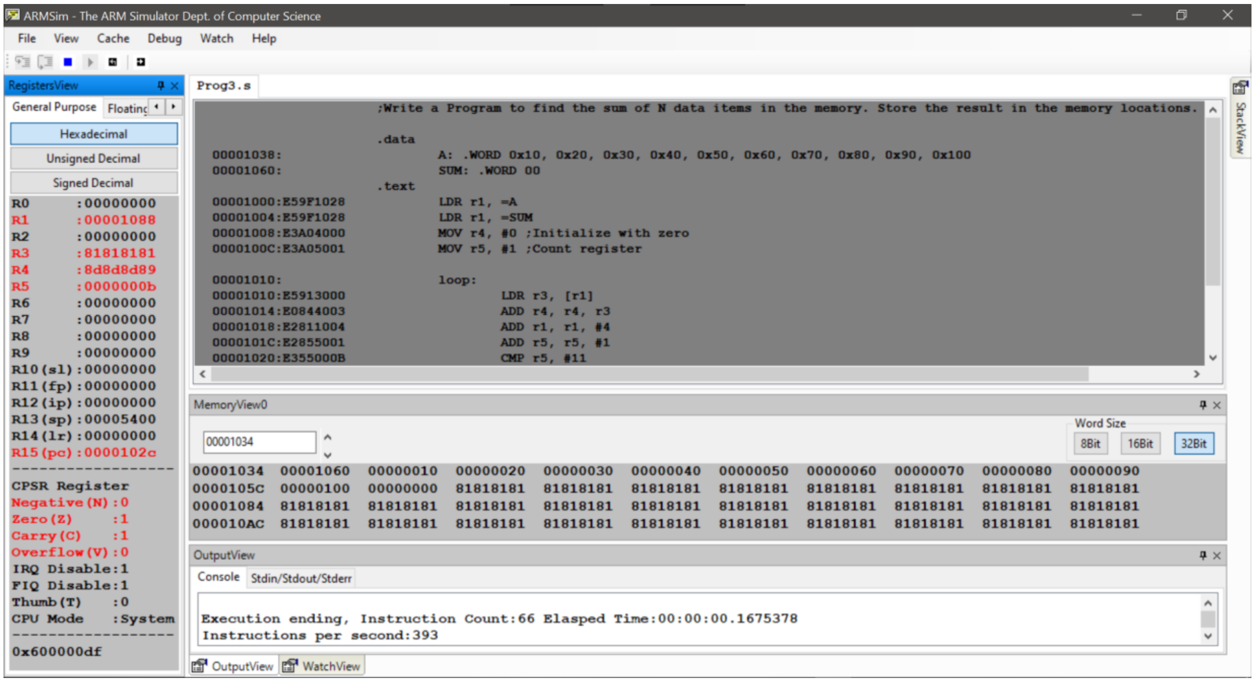
Write a Program to find the sum of N data items in the memory. Store the result in the memory locations. Use Full word.

Program Code

```
.data
        A: .WORD 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80, 0x90,
0x100
        SUM: .WORD 00
.text
        LDR r1, =A
        LDR r2, =SUM
        MOV r4, #0 ;Initialize with zero
        MOV r5, #1 ;Count register

        loop:
                LDR r3, [r1]
                ADD r4, r4, r3
                ADD r1, r1, #4
                ADD r5, r5, #1
                CMP r5, #11
        BNE loop
        STR r4, [r2]
        SWI 0X011
.end
```

Screenshot of ArmSimulator of the Program Executed



```
R0        :00000000
R1        :00001088
R2        :00000000
R3        :81818181
R4        :8d8d8d89
R5        :0000000b
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000102c
------------------
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
```

```
                        ;Write a Program to find the sum of N data items in the memory. Store the result in the memory locations.

                        .data
00001038:                       A: .WORD 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80, 0x90, 0x100
00001060:                       SUM: .WORD 00
                        .text
00001000:E59F1028               LDR r1, =A
00001004:E59F1028               LDR r1, =SUM
00001008:E3A04000               MOV r4, #0 ;Initialize with zero
0000100C:E3A05001               MOV r5, #1 ;Count register

00001010:                       loop:
00001010:E5913000                       LDR r3, [r1]
00001014:E0844003                       ADD r4, r4, r3
00001018:E2811004                       ADD r1, r1, #4
0000101C:E2855001                       ADD r5, r5, #1
00001020:E355000B                       CMP r5, #11
```

```
00001034

00001034  00001060  00000010  00000020  00000030  00000040  00000050  00000060  00000070  00000080  00000090
0000105C  00000100  00000000  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001084  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:66 Elasped Time:00:00:00.1675378
Instructions per second:393
```

Title of the Program

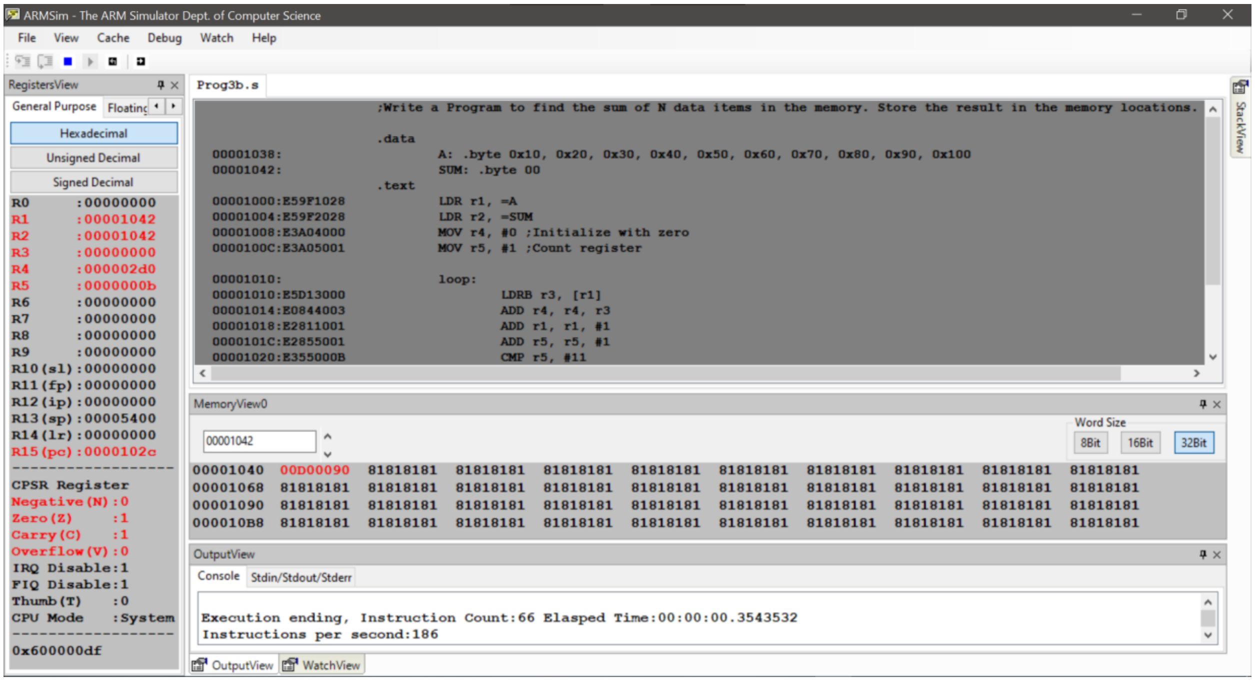
Write a Program to find the sum of N data items in the memory. Store the result in the memory locations. Use BYTE.

Program Code

```
.data
        A: .byte 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80, 0x90,
0x100
        SUM: .byte 00
.text
        LDR r1, =A
        LDR r2, =SUM
        MOV r4, #0 ;Initialize with zero
        MOV r5, #1 ;Count register

        loop:
                LDRB r3, [r1]
                ADD r4, r4, r3
                ADD r1, r1, #1
                ADD r5, r5, #1
                CMP r5, #11
        BNE loop
        STRB r4, [r2]
        SWI 0X011
.end
```

Screenshot of ArmSimulator of the Program Executed



```
R0       :00000000
R1       :00001042
R2       :00001042
R3       :00000000
R4       :000002d0
R5       :0000000b
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00000000
R15(pc)  :0000102c
------------------
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
```

```
                        ;Write a Program to find the sum of N data items in the memory. Store the result in the memory locations.

                        .data
00001038:                       A: .byte 0x10, 0x20, 0x30, 0x40, 0x50, 0x60, 0x70, 0x80, 0x90, 0x100
00001042:                       SUM: .byte 00
                        .text
00001000:E59F1028               LDR r1, =A
00001004:E59F2028               LDR r2, =SUM
00001008:E3A04000               MOV r4, #0 ;Initialize with zero
0000100C:E3A05001               MOV r5, #1 ;Count register

00001010:                       loop:
00001010:E5D13000                       LDRB r3, [r1]
00001014:E0844003                       ADD r4, r4, r3
00001018:E2811001                       ADD r1, r1, #1
0000101C:E2855001                       ADD r5, r5, #1
00001020:E355000B                       CMP r5, #11
```

00001042

```
00001040  00D00090  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001068  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001090  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B8  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:66 Elasped Time:00:00:00.3543532
Instructions per second:186
```

Title of the Program

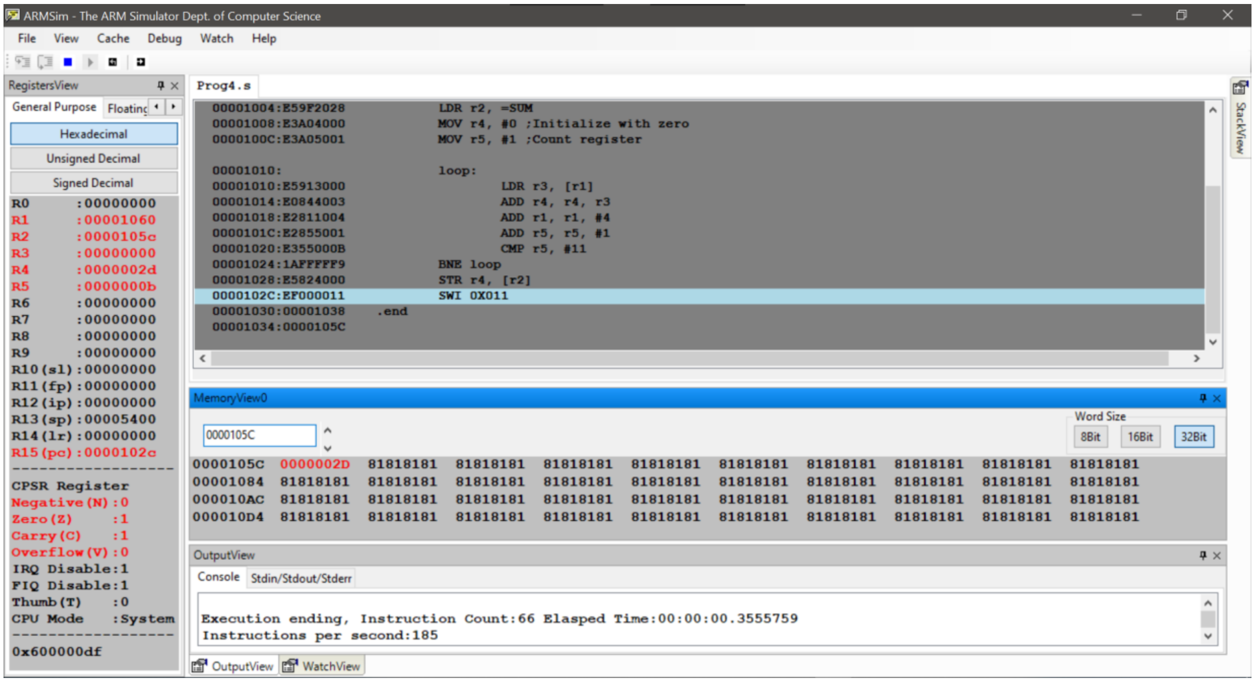
Write a program in ARM7TDMI to find the sum of N natural numbers. Store the result in the memory location. (FULL WORD)

Program Code

```
.data
        A: .WORD 1,2,3,4,5,6,7,8,9
        SUM: .WORD 00
.text
        LDR r1, =A
        LDR r2, =SUM
        MOV r4, #0 ;Initialize with zero
        MOV r5, #1 ;Count register

        loop:
                LDR r3, [r1]
                ADD r4, r4, r3
                ADD r1, r1, #4
                ADD r5, r5, #1
                CMP r5, #11
        BNE loop
        STR r4, [r2]
        SWI 0X011
.end
```

Screenshot of ArmSimulator of the Program Executed



```
00001004:E59F2028            LDR r2, =SUM
00001008:E3A04000            MOV r4, #0 ;Initialize with zero
0000100C:E3A05001            MOV r5, #1 ;Count register

00001010:                    loop:
00001010:E5913000                    LDR r3, [r1]
00001014:E0844003                    ADD r4, r4, r3
00001018:E2811004                    ADD r1, r1, #4
0000101C:E2855001                    ADD r5, r5, #1
00001020:E355000B                    CMP r5, #11
00001024:1AFFFFF9            BNE loop
00001028:E5824000            STR r4, [r2]
0000102C:EF000011            SWI 0X011
00001030:00001038     .end
00001034:0000105C
```

```
R0       :00000000
R1       :00001060
R2       :0000105c
R3       :00000000
R4       :0000002d
R5       :0000000b
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00000000
R15(pc)  :0000102c
------------------
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
```

0000105C

```
0000105c  0000002D  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001084  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010D4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:66 Elasped Time:00:00:00.3555759
Instructions per second:185
```