

Name: Muhammad Nauman Sheikh

Roll No: BSEFI9M504

Computer Organization
&
Assembly Language

Assignment #2

Instructor: Prof Muhammad Arif Butt

①

Task 1

a) $x = 7$
 $y = 12$
 $z = x * y$

Assembly code:

```
@ 7
D = A
@ x
M = D // Assign 7 to x
@ 12
D = A
@ y
M = D // Assign 12 to y
@ 3
M = 0 // Initialize 3 with 0
(Loop) // Label declared
@ x
D = M
@ 3
M = M + D // Add value of x to 3 in each iteration
@ y
M = M - 1 // Decrement value of iterator y by 1
D = M
@ LOOP // Label referred
D: JGT // Jump to LOOP label if iterator is greater than 0
(END)
@ END
0: JMP // Unconditional jump for end
```

②

Machine Instructions

```
0 0 0 0 0 0 0 0 0 0 0 1 1 1
1 1 1 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
1 1 1 0 1 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 1 1 1 1 1 0 0 1 0 0 0 1 0 0 0
1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1
1 1 1 0 1 0 1 0 1 0 0 0 0 0 1 1
```

③

b) $x = 9$
 $y = 3$
 $z = x/y$

Assembly code:

```
@9
D=A
@x
M=D // Assign 9 to x

@3
D=A
@y
M=D // Assign 3 to y

@x
D=M
@temp
M=D // Assign value of x to temp

@3
M=0 // Initialize 3 with 0
// Label declared

@y
D=M
@temp
M=M-D // subtract y from temp and store it in temp

@3
M=M+1 // Increment value of 3 by 1

@temp
D=M
@y
D=D-M
@LOOP
D;JGE // Jump to start of loop if temp >= y

(LOOP)
@END
0;JMP // Unconditional jump for end
```

④

Machine Code:

```

0000 0000 0000 1001
1110 1100 0001 0000
0000 0000 0001 0000
1110 0011 0000 1000
0000 0000 0000 0011
1110 1100 0001 0000
0000 0000 0001 0001
1110 0011 0000 1000
0000 0000 0001 0000
1111 1100 0001 0000
0000 0000 0001 0010
1110 0011 0000 1000
0000 0000 0001 0011
1110 1010 1000 1000
0000 0000 0001 0001
1111 1100 0001 0000
0000 0000 0001 0010
1111 0001 1100 1000
0000 0000 0001 0011
1111 1101 1100 1000
0000 0000 0001 0010
1111 1100 0001 0000
0000 0000 0001 0001
1111 0100 1101 0000
0000 0000 0000 1110
1110 0011 0000 0011
0000 0000 0001 1010
1110 1010 1000 0111

```

(5)

c) $x[5] = \{1, 5, 9, 3, 6\}$
 for ($i=0; i<5; i++$) {
 $x[i] = x[i] + 1;$
 }

Assembly code

```

@20
D=A
@x
M=D // Store address of starting address of array x
@5
D=A
@n
M=D // Store size of array in n
@i
M=0 // iterator variable i=0
@20
M=1 // x[0] = 1
@5
D=A
@21
M=D // x[1] = 5
@9
D=A
@22
M=D // x[2] = 9
@3
D=A
@23
M=D // x[3] = 3
@6
D=A
@24
M=D // x[4] = 6
(LABEL) // Label declared
@i
D=M
@n
D=D-M
@END
D, JEQ // Jump to end if iterator i == n
@x
D=M
@i
A=D+M // Access x[i]
M=M+1 // Increment 1 in x[i]
@i
M=M+1 // Increment i
@LOOP
D, JMP // Label ref'd unconditional jump to start of variable
(ENH) // End declared
@END

```

⑥

0; JMP

Machine Code:

```
0000 0000 0001 0100
1110 1100 0001 0000
0000 0000 0001 0000
1110 0011 0000 1000
0000 0000 0000 0101
1110 1100 0001 0000
0000 0000 0001 0001
1110 0011 0000 1000
0000 0000 0001 0010
1110 1010 1000 1000
0000 0000 0001 0100
1110 1111 1100 1000
0000 0000 0000 0101
1110 1100 0001 0000
0000 0000 0001 0101
1110 0011 0000 1000
0000 0000 0000 1001
1110 1100 0001 0000
0000 0000 0001 0110
1110 0011 0000 1000
0000 0000 0000 0011
1110 1100 0001 0000
0000 0000 0001 0111
1110 0011 0000 1000
0000 0000 0000 0110
1110 1100 0001 0000
0000 0000 0001 1000
1110 0011 0000 1000
0000 0000 0001 0010
1111 1100 0001 0000
0000 0000 0001 0001
1111 0100 1101 0000
0000 0000 0010 1011
1110 0011 0000 0010
0000 0000 0001 0000
1111 1100 0001 0000
0000 0000 0001 0010
1111 0001 1100 1000
0000 0000 0001 1100
1110 1010 1000 0111
0000 0000 0010 1011
1110 1010 1000 0111
```

8

7

d) $a=8$
 $b=9$
 $x=a+b$
 $y=(x+a)*b$

Assembly code:

```

@ 8
D=A
@a
M=D // Assign 8 to a
@9
D=A
@b #A
M=D // Assign 9 to b

@a
D=D+M // Add value of a to the value in D which is equal to b
@x
M=D // Store the new value in x

@a
D=D+M
@temp
M=D // store x+a in a temp variable
@b
D=M
@iter
M=D // run loop for some number of times as value of b
@y
M=0 // initialize y=0
(LABEL) // Label declared
@temp
D=M
@y
M=M+D // add value of temp to y for b number of times
@iter
M=M-1 // decrement iter variable by 1
D=M
@LOOP // label referred
D;JGT // Jump to label loop if iter > 0
(END)
@END
D;JMP // Jump to End label unconditionally

```


⑧

Machine Code:

```

0000 0000 0000 1000
1110 1100 0001 0000
0000 0000 0001 0000
1110 0011 0000 1000
0000 0000 0000 1001
1110 1100 0001 0000
0000 0000 0001 0001
1110 0011 0000 1000
0000 0000 0001 0000
1110 0000 1001 0000
0000 0000 0001 0001
1110 0011 0000 1000
0000 0000 0001 0001
1110 1100 0001 0000
0000 0000 0001 0100
1110 0011 0000 1000
0000 0000 0001 0101
1110 1010 1000 1000
0000 0000 0001 0011
1110 1100 0001 0000
0000 0000 0001 0101
1110 0000 1000 1000
0000 0000 0001 0100
1110 1100 1000 1000
1110 1100 0001 0000
0000 0000 0001 0110
1110 0011 0000 0001
0000 0000 0001 1111
1110 1010 1000 0111

```

9

e) if($x==1$)
 $x=x+1$;
 else
 $x=0$;

Assembly code:

```
@x
D=M // store value of x in Register
D=D-1
@FIRST
D;JEQ // Jump to label First if  $x==1$ 
@x
M=0 // Assign 0 to x if  $x!=1$ 
@END
0;JMP // Jump to end
(FIRST)
@x
M=M+1 // If  $x==1$  then  $x=x+1$ 
(END)
@END
0;JMP // Unconditional jump to end
```

Machine code:

```
0000 0000 0001 0000
1111 1100 0001 0000
1110 0011 1001 0000
0000 0000 0000 1001
1110 0011 0000 0010
0000 0000 0001 0000
1110 1010 1000 1000
0000 0000 0000 1011
1110 1010 1000 0111
0000 0000 0001 0000
1111 1101 1100 1000
0000 0000 0000 1011
1110 1010 1000 0111
```

(10)

Task 2

a) Assembly Code:

@R0

D=M

@R1

D=D-M

@max

M=D

@ASSIGN

D, JGE

@R1

D=M

@max

M=D

@END

O, JMP

(ASSIGN)

@R0

D=M

@max

M=D

(END)

@END

O, JMP

⑪

b) Assembly Code:

```

@20
D=A
@arr
M=D // Store initial address of array
@R1
D=M
@n
M=D // Get value from RAME0 and store it in n
@i
M=0 // initialize i with 0 (iterator)
@20
M=D // arr[0] = 0
@6
D=M
@21
M=D // arr[1] = 6
@8
D=A
@22
M=D // arr[2] = 8
@4
D=A
@23
M=D // arr[3] = 4
@24
M=1 // arr[4] = 1
(LABEL) // Label declared
@i
D=M
@n
D=D-M
@NOT
D.JEQ // If size == i (iterator) jump to location of NOT label
@arr
D=M
@i
A=D+M // Get address of arr[i]
D=M // Store value of arr[i] in D
@R0
D=D-M
@BREAK
D.JEQ // Jump to break if arr[i] == RAME03

```

(12)

```
@i
M=M+1      // increment iterator by 1
@LOOP
0; JMP     // unconditional jump to start
(BREAK)    // label declared in case arr[i]=RAM[0]
@i
D=M
@R3
M=D        // store value of iterator in RAM[3]
@END
0; JMP     // Jump to End
@R3
(NOT)      // label declared in case size == i
@R3
M=-1       // RAM[3] -= 1
(END)
@END
0; JMP     // Unconditional jump to end.
```