Homework-2

1)Assembly code to multiply two unsigned numbers in registers x5 and x6:

Code:

```
.data
.dword   11,35                  #initialized a few values for testing.
.text
lui     x3,0x10000              #initialized register x3 to start address of data segment
ld      x5,0(x3)
ld      x6,8(x3)                #values of operands are stored in separate registers
add   x10,x0,x0                 #Register for answer is initialized to zero
blt     x6,x5,Loop              #If x6 value less than x5 value repeat the Loop instructions x6 times
                                #If x6 value is greater than x5 value swap values
Swap:
    ld   x6,0(x3)
    ld   x5,8(x3)
                                #Loop for repeated addition of first operand, second operand number of times.
Loop:
  beq   x6,x0,Exit
  add   x10,x10,x5
  addi  x6,x6,-1
  beq   x0,x0,Loop
                                #Loop exited
  Exit:
     addi x10,x10,0
```

Reasoning: Given input two numbers,adding the larger value operand smaller value number of times.To run the loop lesser number of times comparison of two operands (and swap to place lower value in x6 register) is done.

2)Equivalent assembly code for given C - code:

```
while (p > q && A[i] != 0) {
                        if (A[i] <= 0)
                        p = p + A[i];
else
                        p = A[i];
                        i = i+2;
 }
```

Assembly code:

```
.data
.dword  11,22,33,76,-54      #initialized sample array values

.text
addi   x11,x0,5              #variable p initialized to 5 and stored in x11
addi   x12,x0,4              #variable q initialized to 4 and stored in x12
lui     x14,0x10000          #x14 is loaded with address of A[0]
add    x13,x0,x14            #x13 register to update variable i using it's address
```

```
Loop:
      ld     x15,0(x13)              #x15 register value refers to A[i]
      beq    x15,x0,Exit             #If A[i]==0 or p<=q exit loop
      bge    x12,x11,Exit
      blt    x0,x15,Else             #If A[i]>0 exit loop
      add    x11,x11,x15             #p=p+A[i]
      addi   x13,x13,16              #i=i+2 (2*8 bytes to be incremented)
      beq    x0,x0,Loop
Else:
      add    x11,x0,x15              #p=A[i]
      addi   x13,x13,16              #i=i+2 (2*8 bytes to be incremented)
      beq    x0,x0,Loop
Exit:
      addi   x10,x10,0
```

Reasoning: While loop,if condition executed in assembly using branching instructions
beq x0,x0,Loop is responsible for continuous running of the loop until Else or Exit label is entered.
blt,bge instructions used where comparison of two values needed.

3)Given two double words in data segment:
0xa55aa5a593933939
0x39933939a55aa5a5
Their byte-wise memory arrangement according to LE format:

Continuous memory allocation inside data segment represented as byte-0 to byte-15(originally stored from byte-0 to byte-7 is continued by byte-0 to byte-7 of other number):

Byte-0	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7
0x39	0x39	0x93	0x93	0xa5	0xa5	0x5a	0xa5

Byte-8	Byte-9	Byte-10	Byte-11	Byte-12	Byte-13	Byte-14	Byte-15
0xa5	0xa5	0x5a	0xa5	0x39	0x39	0x93	0x39

Instruction	x3 value	Explanation
lhu x3,0(x1)	0x0000000000003939	half-word(2 bytes) loaded from byte-0 to byte-1
lh x3,0(x1)	0x0000000000003939	sign-extended half-word byte-0 to byte-1

lh x3,2(x1)	0xffffffffffff9393	half-word(2 bytes) loaded from byte-2 to byte-3
ld x3,0(x1)	0xa55aa5a593933939	double-word(8 bytes) loaded from byte-0 to byte-7
lw x3,12(x1)	0x0000000039933939	word(4 bytes) loaded from byte-12 to byte-15
lbu x3,7(x1)	0x00000000000000a5	1 byte loaded from byte-7.
lb x3,7(x1)	0xffffffffffffffa5	sign-extended 1 byte loaded from byte-7.
lb x3,6(x1)	0x000000000000005a	1 byte loaded from byte-6 of x1.
ld x3,3(x1)	0x5aa5a5a55aa5a593	double-word (8 bytes) loaded from byte-3 to byte-10
lwu x3,6(x1)	0x00000000a5a5a55a	word(4 bytes) loaded from byte-6 to byte-9

Every value from the first byte to the last byte is represented from MSB to LSB in register x3.