Project – 1

Name: Uros Nikolic

Pledge: I pledge my honor that I have abided by Stevens Honor system.

Create Histogram:

```
_start:
    LDR x0, =inpdata                // Intalization of the first element of array
    LDR x1, =inplen                 // Intalization of the length of the array
    LDR x27, [x1]                   // x27 = loading the value of the length of the array
    LDR X2, =histogram              // x2 = loading the address of the first element of histogram array
    MOV X3, 100                     // x3 = loading the value of the length of the histogram array
    B createHistogram               // Unconditional branc to createHistogram to call the procedure
    MOV w8, #93                     // x8 = loading the value of the system call
    MOV x0, #0                      // x0 = loading the value of the exit status
    svc #0                          // system call to exit the program

createHistogram:
    MOV x11, #0                     // x11 = loading the value of the counter
    B loopOfI                       // Unconditional branch to loopOfI to start the loop


loopOfI:
    CMP x11, x27                    // if(counter < arraylength)
    B.gt print                      // if(counter >= arraylength) go directly to print
    LDR x20, [x0, x11, LSL #3]      // Shifting to left 3 times, then add the index to the address of the array
    LDR x21, [x2, x20, LSL #3]      // Shifitng to left 3 times, then add the index to the address of the histogram array
    ADD x21, x21, #1                // Increment the value of the histogram array
    STR x21, [x2, x20, LSL #3]      // Store the value of the histogram array
    ADD x11, x11, #1                // Increment the counter by 1
    B loopOfI                       // Unconditional branch to loopOfI to continue the loop
```

The Function createHistogram loops through all the items of input array and increments the Counterarray[i] where each i-th element is the counter for each number of input array elements. We loop through all elements of input array, then we branch to print label if there is no elements of input array.

Print for createHistogram and calling the rank procedure

```
print:
    LDR x0, =outLabels              // x0 = loading the string for printout: "Number   Count"
    Bl printf                       // Call the printf function to print the string
    MOV x26, #0                     // x26 = loading the value of the counter
    B loop                          // Unconditional branch to loop to start the loop

loop:
    CMP x26, x27                    // if(counter < arraylength)
    B.gt end1                       // if(counter >= arraylength) go directly to end1
    LDR x0, =outdata                // x0 = loading the string for printout: "%d\t%d\n"
    MOV x1, x26                     // x1 = loading the value of the counter
    LDR x2, =histogram              // x2 = loading the address of the first element of histogram array
    LDR x2, [x2, x26, LSL #3]       // Shifting to left 3 times, then add the index to the address of the histogram array
    Bl printf                       // Call the printf function to print the string
    ADD x26, x26, #1                // Increment the counter by 1
    B loop                          // Unconditional branch to loop to continue the loop

end1:
    MOV x29, #0                     // x29 = setting the last element of the histogram array to 0
    SUB x29, x29, #1                // x29 = setting the last element of the histogram array to -1
    LDR x2, =histogram              // x2 = loading the address of the first element of histogram array
    STR x29, [x2, x26, LSL #3]      // Shifting to left 3 times, then add the index to the address of the histogram array
    LDR x0, =rank                   // x0 = loading the address of the rank
    LDR x0, [x0]                    // x0 = loading the value of the rank
    LDR x1, =histogram              // x1 = loading the address of the first element of histogram array
    BL rankfunc                     // Call the rankfunc function to find the rank
    MOV x29, x0                     // Saves the rank value in x29
    LDR x0, =outRank                // x0 = loading the string for printout: "The value of the rank-%d element is %d\n"
    LDR x1, =rank                   // x1 = loading the address of the rank
    LDR x1, [x1]                    // x1 = loading the value of the rank
    MOV x2, x29                     // x2 = loading the exact value for the rank asked from x29
    BL printf                       // Call the printf function to print the string
```

First, we print the Labels as requested in the output format, then we branch and link with print function to print our labels. After printout of labels, we are branching to loop which will start a print procedure for the results found in the create histogram loopOfI part. In We print i-th element of input string alongside i-th element of array of counters. In the end1 part we are printing all the values, and we are calling the function rankfunc.

Rank Function:

```
.func rankfunc
rankfunc:
    str x30, [sp]                   // Calling convention to set up the stack pointer to x30
    mov x15, x0                     // x15 = loading the value of the rank
    MOV x11, #0                     // Initializing the index to 0
    MOV x12, #0                     // Initializing the counter to 0
l1:
    LDR x3, [x1, x11, LSL #3]       // Shifting to left 3 times, then add the index to the address of the rank
    CMP x3, x29                     // if(rank == histogram[index])
    b.eq break                      // if(rank == histogram[index]) go directly to break
    ADD x12, x12, X3                // Increment the counter by the value of the histogram array
    CMP x12, x15                    // if(counter > rank)
    b.ge break                      // if(counter > rank) go directly to break
    ADD x11, x11, #1                // Increment the index by 1
    B l1                            // Unconditional branch to l1 to continue the loop

break:
    MOV x0, x11                     // x0 = loading the value of the index
    ldr x30, [sp]                   // Calling convention to set up the stack pointer to x30
    br x30                          // Unconditional branch to x30 to return to the main function

.endfunc
```

Rankfunc is supposed to go through all the elements of array of counters and to find the input rank which will print the number who is the rank for input number of the rank.