**Lab Manual – Nested Loops + Conditional Jumps**

**Activity 1 [Conditional Jump]:** Find frequency of an integer from an array of 10 integers.

For example frequency of 2 in array 2, 4, 2, 8, 2 is 3.

**Activity 2 [Nested Loop]:** Write a program that computes factorial of a number without using multiplication instruction.

**Activity 3:** Write a program that compresses an ordered array (having multiple occurrences of one integer). You are not allowed to use any extra array, just modify the input array.

**Sample Run:**

|  |
| --- |
| **Arr:** 2,2,2,3,4,4,5,5,5,6  After Compression, **Arr:** 2,3,4,5,6,0,0,0,0,0 |

**Activity 4:** We did following Sorting Example in class.

1. Run this code (without any modification) on signed data and verify output.
2. Modify this code to sort **signed data** and test it.

|  |
| --- |
| ; sorting a list of ten numbers using bubble sort  [org 0x0100]  jmp start  data: dw 6, 5, 3, 1, 2  swap: db 0  start: mov cx, 10 ; n = (5x2)-2  outerloop: mov bx, data ; initialize start ptr  sub cx, 2  mov di, bx  add di, cx ; endptr = startptr + n  mov byte[swap], 0 ; rest swap flag to no swaps  innerloop: mov ax, [bx] ; load number in ax  cmp ax, [bx+2] ; compare with next number  jbe noswap ; no swap if already in order    mov dx, [bx+2] ; load second element in dx  mov [bx+2], ax ; store first number in second  mov [bx], dx ; store second number in first  mov byte [swap], 1 ; flag that a swap has been done  noswap: add bx, 2 ; advance bx to next index  cmp bx, di ; are we at last index  jne innerloop ; if not compare next two  cmp byte [swap], 1 ; check if a swap has been done  je outerloop ; if yes make another pass  mov ax, 0x4c00 ; terminate program  int 0x21 |

**Practice Problems**

1. Find min and max elements from an array of 10 integers. Write two functions for
2. Unsigned Numbers
3. Signed Numbers
4. Write a program that finds union of two sets (ordered integer arrays). Assume there will be at max 10 elements in union.

**Sample Run:** (Zero indicates end of array)

|  |
| --- |
| Set1: 1,4,6,0  Set2: 1,3,5,8,0  Union: 1, 3, 4, 5, 6, 8, 0 |

1. Write a program that finds intersection of two sets (ordered integer arrays). Assume there will be at max 10 elements in union.

**Sample Run:** (Zero indicates end of array)

|  |
| --- |
| Set1: 1,4,6,0  Set2: 1,3,6,8,0  Intersection: 1, 6, 0 |

1. Write a program that finds staring index of a subset from a set (smaller array from a larger array).

**Sample Run:**

|  |
| --- |
| Set:1,2,4,1,1,2,3,1,2,5  Subset: 1,2,3  **Index: 4 (After program. -1 if subset not found.)** |