**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **23/6/2020** | | | | | **Name:** | **Banish M G** | |
| **Sem & Sec** | **8th Sem** | | | | | **USN:** | **4AL16CS20** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SMS** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **NOT RECIEVED** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Python for data science** | | | | | | | |
| **Certificate Provider** | | | **udemy** | | **Duration** | | | **23hrs** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  Write a C Program to Sort a stack using a temporary stack | | | | | | | | |
| **Status:COMPLETED** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Banish\_MG** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)



Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

def sortStack ( stack ):

    tmpStack = createStack()

    while(isEmpty(stack) == False):

        tmp = top(stack)

        pop(stack)

        while(isEmpty(tmpStack) == False and

             int(top(tmpStack)) > int(tmp)):

            push(stack,top(tmpStack))

            pop(tmpStack)

        push(tmpStack,tmp)

    return tmpStack

def createStack():

    stack = []

    return stack

def isEmpty( stack ):

    return len(stack) == 0

def push( stack, item ):

    stack.append( item )

def top( stack ):

    p = len(stack)

    return stack[p-1]

def pop( stack ):

    # If stack is empty

    # then error

    if(isEmpty( stack )):

        print("Stack Underflow ")

        exit(1)

    return stack.pop()

def prints(stack):

    for i in range(len(stack)-1, -1, -1):

        print(stack[i], end = ' ')

    print()

stack = createStack()

push( stack, str(34) )

push( stack, str(3) )

push( stack, str(31) )

push( stack, str(98) )

push( stack, str(92) )

push( stack, str(23) )

print("Sorted numbers are: ")

sortedst = sortStack ( stack )

prints(sortedst)