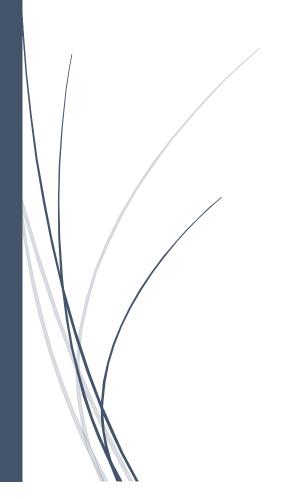
4/1/2023

# COAL ASSIGNMENT 2

20k1898 Muhammad BILAL KHAN



#### QUESTION # 1:

```
Include Irvine32.inc
.data
arr SDWORD 6 DUP(?) j
SDWORD ? k SDWORD ?
.code
     MAIN PROC
                           mov
ecx, LENGTHOF arr
                           mov
ebx,0
                getInput:
                mWrite " Enter value # "
           mov eax,ebx
                                       inc
eax
               call WriteDec
     mWrite " : "
call ReadInt
                mov [arr+ebx*4],eax
                inc ebx
     loop getInput
call getJK
                      push
OFFSET arr
                      push
SIZEOF arr
                      push
           push k
           call arraySum
call printResults
call getJK
                      push
OFFSET arr
                      push
SIZEOF arr
                      push j
     push k
           call arraySum
call printResults
           exit
     MAIN ENDP
     printResults PROC
```

```
mWrite "The sum of all the elements in the range: "
     call WriteInt
           call crlf
           ret
     printResults ENDP
     getJK PROC
           mWrite "Enter the value of j: "
     call ReadInt
           mov j,eax
           mWrite "Enter the value of k: "
     call ReadInt
                            mov k,eax
           ret
     getJK ENDP
     arraySum PROC uses EBX ECX EDX ESI
                                                   local
first:SDWORD,last:SDWORD,sizeArray:DWORD
           mov esi, [ebp + 20]
     mov eax, [ebp + 16]
mov sizeArray,eax
mov eax, [ebp + 12]
mov first, eax
                      mov
eax,[ebp + 8]
                      mov
last,eax
                mov eax,0
     mov edx,0
mov ecx,sizeArray
sumInRange:
                mov ebx, [esi + edx * 4]
           cmp ebx, first
jge checkIfWithinRange
jmp continueLoop
checkIfWithinRange:
cmp ebx,last
                                  jle
addIt
                      jmp continueLoop
           addIt:
add eax, ebx
continueLoop:
```

inc ecx
sub ecx,4
inc edx
loop sumInRange
ret 16
arraySum ENDP
END MAIN

#### QUESTION # 2:

```
Include Irvine32.inc
.data
          arr DWORD 5
DUP(?)
.code
     MAIN PROC
                           mov
ecx,LENGTHOF arr
                           mov
ebx,0
                getInput:
                mWrite " Enter element # "
           mov eax,ebx
                                      inc
eax
                call WriteDec
     mWrite " : "
call ReadInt
                           mov
[arr+ebx*4],eax
                inc ebx
                                 loop
                call selectionSort
getInput
mov ecx, LENGTHOF arr
                                 mov
ebx,0
                printSortedArray:
     mov eax,[arr + ebx * 4]
                           mWrite " "
call WriteDec
           inc ebx
           loop printSortedArray
           exit
```

```
MAIN ENDP
     SWAP PROC
           push ebp
mov ebp,esp
                       mov
edx,[ebp + 8]
                       push
edx
           mov eax, [arr + edx * 4]
     mov edx, [ebp + 12]
xchg eax, [arr + edx * 4]
           pop edx
           mov [arr + edx * 4], eax
           pop ebp
ret 8
     SWAP ENDP
     selectionSort PROC
           LOCAL largest:DWORD,i:DWORD,j:DWORD
     mov ecx, LENGTHOF arr
largest,0
                       dec ecx
                                        mov
i,ecx
                 mov j,ecx
outerLoop:
                            mov ebx,i
     mov largest, ebx
                                        push ecx
           mov edx,i
                                        mov
j,edx
                       innerLoop:
     dec j
                       mov edx,j
                      mov eax, [arr + edx * 4]
                 mov edx, largest
     mov ebx, [arr + edx * 4]
cmp eax,ebx
                                  jg markNewMax
                 jmp continueLoop
     markNewMax:
```

mov

push i

continueLoop:

mov edx,j

largest,edx

loop innerLoop

call SWAP

push largest

### QUESTION # 3:

```
Include Irvine32.inc
.data
          arr BYTE 10
DUP(?)
.code
     MAIN PROC
                           mov
ecx,LENGTHOF arr
                           mov
ebx,0
                getInput:
                mWrite "Enter value # "
          mov eax,ebx
inc eax
               call WriteDec
     mWrite " : "
call ReadInt
                         mov [arr
+ ebx],al
```

```
inc ebx
loop getInput
                      mov
esi,OFFSET arr
                      mov
ebx,LENGTHOF arr
                            call
bubbleSort
                      mov ebx,0
           mov ecx, LENGTHOF arr
     printArray:
mov al,[arr + ebx]
call WriteDec
                            inc ebx
           mWrite " "
           loop printArray
           exit
     MAIN ENDP
     bubbleSort PROC
           mov edi,esi
           mov ecx, ebx
           dec ecx
           mov ebx,0
           mov eax,0
           outerLoop:
            push ecx
                 mov esi,edi
innerLoop:
                                 mov
al,[esi] mov bl,[esi+ 1]
cmp al,bl jg swapElements
     continueLoop:
                            inc esi
                 loop innerLoop
     pop ecx
           loop outerLoop
jmp endProgram
swapElements:
                            mov
al,[esi]
                      mov bl,[esi
                      xchg al,bl
+ 1]
           mov [esi],al
mov [esi + 1],bl
```

jmp continueLoop
endProgram: ret 8
 bubbleSort ENDP
END MAIN

## QUESTION # 4:

```
exit
MAIN ENDP

factorial PROC
mov eax,1
cmp N,0
jle endProgram
mov ecx,N
calculate:
mov edx,0
mul ecx
loop calculate
endProgram:
ret
factorial ENDP
END MAIN
```