# Name: Muhammad Maiz Nadeem

Reg. ID: SP21-BCS-052

# Recursion 2

# Question 1:

Implement the following recursive algorithm on array:

A DW 78, 24, 45, 87

N DW 5 (Size of Array)

i DW 0 (Local Counter)

Minimum = FindMin (i) { // means that the functions

if (i == N-1) { returns answer in minimum.

return A[i]

}

else {

Minimum = FindMin (i+1)

Return (MIN (Minimum, A[i]) ) ) // return which ever is minimum of

} the two parameters.

}

# Answer:

|  |
| --- |
| ORG 100h |
|  |
| .DATA |
|  |
| A       *DW*  5, 4, 3, 2, 1       ; (Global) Array that holds terms |
| I       *DW*  0                   ; (Local)  Array's index |
| N       *DW*  5                   ; (Global) Number of terms remaining |
| MIN     *DW*  0                   ; (Global) Holds minimum value |
|  |
| .CODE |
|  |
| MAIN PROC |
|  |
| LEA SI, A |
| PUSH 0 |
| CALL FINDMIN |
|  |
| RET |
| MAIN ENDP |
|  |
|  |
| FINDMIN PROC |
|  |
| MOV BP, SP |
|  |
| MOV BX, [BP+2] |
| MOV DX, N |
| ADD DX, N |
| SUB DX, 2 |
| CMP BX, DX |
| JNE AGAIN |
|  |
| MOV BX, [BP+2] |
| MOV DX, [SI+BX] |
| MOV MIN, DX |
| JMP EXIT |
|  |
| AGAIN:  ADD I, 2 |
| PUSH I |
| CALL FINDMIN |
|  |
| MOV BP, SP |
| MOV BX, [BP+2] |
| MOV AX, [SI+BX] |
| CMP MIN, AX |
| JLE EXIT |
| MOV MIN, AX |
|  |
|  |
| EXIT: |
| RET 2 |
|  |
| FINDMIN ENDP |

# Output:

Graphical user interface, application

Description automatically generated