Latihan 10

1. Tuliskan kembali algoritma pengurutan apung sedemikian sehingga elemen- elemen terurut “tumbuh” dari “kanan” ke “kiri” (atau dari “bawah” ke “atas”).

Jawab :

Function getPengurutanApung (input L : LarikInt, input n : integer ) **🡪** LarikInt

Deklarasi

i,k,temp : integer

Algoritma

for i 🡨 n downto 1 do

for k 🡨 1 to n do

if (L[k]<L[k+1]) then

temp 🡨L[k]

L[k] 🡨 L[k+1]

L[k+1] 🡨 temp

endif

endfor

endfor

return L

1. Modifikasi algoritma pengurutan sisip untuk mengurutkan data yang di- entry dari papan ketik. Dalam hal ini, setiap kali data dibaca, data tersebut dicarikan posisinya yag tepat di dalam larik sehinnga larik tetap terurut menaik

Jawab :

Function getPengurutanSisip (input L : LarikInt, input n : integer) **🡪** LarikInt

Deklarasi

i, j, k, y : integer

ketemu : boolean

Algoritma

for k 🡨 n downto 1 do

read L[k]

for i 🡨 2 to n do

y 🡨 L[i]

j 🡨 j -1

ketemu 🡨 false

while (j>=0) && (not ketemu) do

if (y < L[j]) then

L[j+1] 🡨 L[j]

j 🡨 j -1

else

ketemu 🡨 true

endif

endwhile

L[j+1] 🡨 y

endfor

write L

endfor

return L

1. Buat notasi algoritmik dalam bentuk pseudo-code untuk algoritma pengurutan larik multidimensi!

Jawab :

Function getLarikMultidimensi (input L : MatriksInt, input nbar, nkol : integer) **🡪** MatriksInt

Deklarasi

i, j, temp, tambah,ntotal : integer

Array : array [1… n] of integer

Algoritma

Ntotal 🡨 nbar\*nkol

tambah 🡨 0

for i 🡨 1 to nbar do

for j 🡨 1 to nkol do

Array[tambah] 🡨 L[i][j]

tambah 🡨 tambah + 1

endfor

endfor

for i 🡨 1 to ntotal do

for j 🡨 ntotal downto i do

if (Array[j] < Array[j-1]) then

temp 🡨Array[j]

Array[j] 🡨 Array[j-1]

L[j-1] 🡨 temp

endif

endfor

endfor

tambah 🡨 0

for i 🡨 1 to nbar do

for j 🡨 1 to nkol do

L[i][j] 🡨 Array[tambah]

tambah 🡨 tambah + 1

endfor

endfor

return L