Baskoro Adi Wicaksono

18219113

Sebelumnya, saya menggunakan dua tipe data dengan tipe data integer untuk mendefinisikan indeks table dan elemen yang terkandung di tabel, serta 1 variabel dengan tipe data bentukan untuk mendeklarasikan list/array.

type IdxType: integer { tipe indeks tabel }

type ElType: integer { elemen tabel }

type List: < A: array [0..MaxEl-1] of ElType > { penyimpanan elemen List. }

Soal 1

function IsSimetrik(L: List) → boolean

{ Menghasilkan true jika list simetrik

List disebut simetrik jika:

elemen pertama = elemen terakhir,

elemen kedua = elemen sebelum terakhir, dan seterusnya

List kosong adalah list simetrik }

{KAMUS}  
i, Nmax : integer

L : array [1..Nmax] of ElType

{ALGORITMA}

if L[i] ≠{}

while i < Nmax/2 do

if (L[i] == L[Nmax – (i-1)]

return True

i += 1

else

return False

else

return True

Soal 2

function PlusList(L1,L2: List) → List

{ Prekondisi: L1 dan L2 berukuran sama dan tidak kosong

Mengirimkan L1+L2, yaitu penjumlahan setiap elemen L1

dan L2 pada indeks yang sama }

{KAMUS}

L1,L2, L3 : array [1..Nmax] of ElType

i, Nmax : integer

{ALGORITMA}

while i < n do

L3.A[i] = L1.A[i] + L2.A[i]

i += 1

return L3

Soal 3

function IsEQ(L1,L2: List) → boolean

{ Mengirimkan true jika L1=L2, yaitu jika ukuran L1=L2

dan semua elemen L1 dan L2 pada indeks yang sama bernilai sama }

{KAMUS}

L1 : array[1..Nmax1] of ElType

L2 : array[1..Nmax2] of ElType

i, Nmax1, Nmax2 : integer

temp : boolean

{ALGORITMA}

i ← 0

temp ← False

if Nmax1 == Nmax2

while i < Nmax1 do

if L1[i] == L2[i]

temp ← True

else

temp ← False

return temp

else

return temp

Soal 4

function Search(L: List, X: ElType) → IdxType

{ Search apakah ada elemen list L yang bernilai X.

Jika ada, menghasilkan indeks i terkecil, dengan elemen ke-i = X.

Jika tidak ada, mengirimkan InvalidIdx.

Menghasilkan indeks tak terdefinisi (InvalidIdx) jika list kosong.

Memakai skema searching tanpa Boolean. }

{KAMUS}

L : array[1..Nmax] of int

i, Nmax, X, temp : integer

{ALGORITMA}

input(X)

temp ← InvalidIdx

if L == {}

return temp

else

while i < Nmax do

if L[i] == X

temp ← i

else

i += 1

return temp

Soal 5

function CountX(L: List, X: ElType) → integer

{ Menghasilkan berapa banyak kemunculan elemen bernilai X di List L.

Jika list kosong, menghasilkan 0 }

{KAMUS}

L : array[1..Nmax] of ElType

i, Nmax, X, temp : integer

{ALGORITMA}

Input(X)

temp ← 0

If L == {}

temp ← 0

else

i traversal[1..Nmax]

if (L[i] == X)

temp ← temp + 1

return temp

Soal 6

procedure InsertUniq(input/output L: List, input X: ElType)

{ Menambahkan X sebagai elemen terakhir list, pada list dengan elemen unik }

{ I.S. List L boleh kosong, tetapi tidak penuh }

{ dan semua elemennya bernilai unik, tidak terurut }

{ F.S. Menambahkan X sebagai elemen terakhir L, }

{ jika belum ada elemen yang bernilai X. }

{ Jika sudah ada elemen list yang bernilai X }

{ maka I.S. = F.S. }

{KAMUS}  
L : array [1..Nmax] of ElType

X, i, j, Nmax, Nisi, Mark : integer

temp : boolean

{ALGORITMA}

Nisi ← 0

temp ← False

while L[i] ≠ Mark

Nisi ← Nisi + 1

i ← i + 1

i ← 0

input(X)

while i < Nisi

if L[i] ≠ X

temp ← True

i ← i + 1

if temp = True

Nmax ← Nisi + 1

L[Nmax] ← X

return