*#include<stdio.h>*

*#include "array.h"*

*/\* \*\*\*\*\*\*\*\*\*\* KONSTRUKTOR \*\*\*\*\*\*\*\*\*\* \*/*

*/\* Konstruktor : create tabel kosong \*/*

*void MakeEmpty (TabInt \*T){*

*(\*T).Neff = 0;*

*};*

*/\* I.S. sembarang \*/*

*/\* F.S. Terbentuk tabel T kosong dengan kapasitas IdxMax-IdxMin+1 \*/*

*/\* \*\*\*\*\*\*\*\*\*\* SELEKTOR \*\*\*\*\*\*\*\*\*\* \*/*

*/\* \*\*\* Banyaknya elemen \*\*\* \*/*

*int NbElmt (TabInt T){*

*return T.Neff;*

*};*

*/\* Mengirimkan banyaknya elemen efektif tabel \*/*

*/\* Mengirimkan nol jika tabel kosong \*/*

*/\* \*\*\* Daya tampung container \*\*\* \*/*

*int MaxNbEl (TabInt T){*

*return IdxMax;*

*};*

*/\* Mengirimkan maksimum elemen yang dapat ditampung oleh tabel \*/*

*/\* \*\*\* Selektor INDEKS \*\*\* \*/*

*IdxType GetFirstIdx (TabInt T){*

*return IdxMin;*

*};*

*/\* Prekondisi : Tabel T tidak kosong \*/*

*/\* Mengirimkan indeks elemen pertama \*/*

*IdxType GetLastIdx (TabInt T){*

*return NbElmt(T);*

*};*

*/\* Prekondisi : Tabel T tidak kosong \*/*

*/\* Mengirimkan indeks elemen terakhir \*/*

*/\* \*\*\* Menghasilkan sebuah elemen \*\*\* \*/*

*ElType GetElmt (TabInt T, IdxType i){*

*return T.TI[i];*

*};*

*/\* Prekondisi : Tabel tidak kosong, i antara FirstIdx(T)..LastIdx(T) \*/*

*/\* Mengirimkan elemen tabel yang ke-i \*/*

*/\* \*\*\* Selektor SET : Mengubah nilai TABEL dan elemen tabel \*\*\* \*/*

*/\* Untuk type private/limited private pada bahasa tertentu \*/*

*void SetTab (TabInt Tin, TabInt \*Tout){*

*int i ;*

*for(i = IdxMin;i<=IdxMax;i++){*

*(\*Tout).TI[i] = Tin.TI[i];*

*}*

*(\*Tout).Neff = Tin.Neff;*

*};*

*/\* I.S. Tin terdefinisi, sembarang \*/*

*/\* F.S. Tout berisi salinan Tin \*/*

*/\* Assignment THsl -> Tin \*/*

*void SetEl (TabInt \*T, IdxType i, ElType v){*

*(\*T).TI[i] = v;*

*if (i == GetLastIdx(\*T) + 1)*

*{*

*(\*T).Neff ++;*

*}*

*};*

*/\* I.S. T terdefinisi, sembarang \*/*

*/\* F.S. Elemen T yang ke-i bernilai v \*/*

*/\* Mengeset nilai elemen tabel yang ke-i sehingga bernilai v \*/*

*void SetNeff (TabInt \*T, IdxType N){*

*(\*T).Neff = N;*

*};*

*/\* I.S. T terdefinisi, sembarang \*/*

*/\* F.S. Nilai indeks efektif T bernilai N \*/*

*/\* Mengeset nilai indeks elemen efektif sehingga bernilai N \*/*

*/\* \*\*\*\*\*\*\*\*\*\* Test Indeks yang valid \*\*\*\*\*\*\*\*\*\* \*/*

*boolean IsIdxValid (TabInt T, IdxType i){*

*return ((i >= IdxMin) && (i <= IdxMax));*

*};*

*/\* Prekondisi : i sembarang \*/*

*/\* Mengirimkan true jika i adalah indeks yang valid utk ukuran tabel \*/*

*/\* yaitu antara indeks yang terdefinisi utk container\*/*

*boolean IsIdxEff (TabInt T, IdxType i){*

*return ((i >= GetFirstIdx(T)) && (i <= GetLastIdx(T)));*

*};*

*/\* Prekondisi : i sembarang\*/*

*/\* Mengirimkan true jika i adalah indeks yang terdefinisi utk tabel \*/*

*/\* yaitu antara FirstIdx(T)..LastIdx(T) \*/*

*/\* \*\*\*\*\*\*\*\*\*\* TEST KOSONG/PENUH \*\*\*\*\*\*\*\*\*\* \*/*

*/\* \*\*\* Test tabel kosong \*\*\* \*/*

*boolean IsEmpty (TabInt T){*

*return (NbElmt(T) == 0);*

*};*

*/\* Mengirimkan true jika tabel T kosong, mengirimkan false jika tidak \*/*

*/\* \*\*\* Test tabel penuh \*\*\* \*/*

*boolean IsFull (TabInt T){*

*return (NbElmt(T) == MaxNbEl(T));*

*};*

*/\* Mengirimkan true jika tabel T penuh, mengirimkan false jika tidak \*/*

*/\* \*\*\*\*\*\*\*\*\*\* BACA dan TULIS dengan INPUT/OUTPUT device \*\*\*\*\*\*\*\*\*\* \*/*

*void TulisIsi (TabInt T){*

*int i;*

*if(IsEmpty(T)){*

*printf("Tabel kosong\n");*

*}*

*else{*

*for(i=GetFirstIdx(T);i<=GetLastIdx(T);i++){*

*printf("%d:%d\n",i,T.TI[i]);*

*}*

*}*

*};*

*/\* Proses : Menuliskan isi tabel dengan traversal \*/*

*/\* I.S. T boleh kosong \*/*

*/\* F.S. Jika T tidak kosong : indeks dan elemen tabel ditulis berderet ke bawah \*/*

*/\* Jika isi tabel [1,2,3] maka akan diprint*

*0:1*

*1:2*

*2:3*

*\*/*

*/\* Jika T kosong : Hanya menulis "Tabel kosong" \*/*

*/\* \*\*\*\*\*\*\*\*\*\* OPERATOR ARITMATIKA \*\*\*\*\*\*\*\*\*\* \*/*

*/\* \*\*\* Aritmatika tabel : Penjumlahan, pengurangan, perkalian, ... \*\*\* \*/*

*TabInt PlusTab (TabInt T1, TabInt T2){*

*int i;*

*TabInt THsl;*

*MakeEmpty(&THsl);*

*THsl.Neff = T1.Neff;*

*for(i=GetFirstIdx(T1);i<=GetLastIdx(T1);i++){*

*SetEl(&THsl, i, GetElmt(T1, i) + GetElmt(T2, i));*

*}*

*return(THsl);*

*};*

*/\* Prekondisi : T1 dan T2 berukuran sama dan tidak kosong \*/*

*/\* Mengirimkan T1 + T2 \*/*

*TabInt MinusTab (TabInt T1, TabInt T2){*

*int i;*

*TabInt THsl;*

*MakeEmpty(&THsl);*

*THsl.Neff = T1.Neff;*

*for(i=GetFirstIdx(T1);i<=GetLastIdx(T1);i++){*

*SetEl(&THsl, i, GetElmt(T1, i) - GetElmt(T2, i));*

*}*

*return(THsl);*

*};*

*/\* Prekondisi : T1 dan T2 berukuran sama dan tidak kosong \*/*

*/\* Mengirimkan T1 - T2 \*/*

*/\* \*\*\*\*\*\*\*\*\*\* NILAI EKSTREM \*\*\*\*\*\*\*\*\*\* \*/*

*ElType ValMax (TabInt T){*

*int i;*

*ElType max;*

*max = T.TI[GetFirstIdx(T)];*

*for (i = (GetFirstIdx(T)+1); i <= (GetLastIdx(T)); i++)*

*{*

*if (max < T.TI[i])*

*{*

*max = T.TI[i];*

*}*

*}*

*return(max);*

*};*

*/\* Prekondisi : Tabel T tidak kosong \*/*

*/\* Mengirimkan nilai maksimum tabel \*/*

*ElType ValMin (TabInt T){*

*int i;*

*ElType min;*

*min = T.TI[GetFirstIdx(T)];*

*for (i = (GetFirstIdx(T)+1); i <= (GetLastIdx(T)); i++)*

*{*

*if (min > T.TI[i])*

*{*

*min = T.TI[i];*

*}*

*}*

*return(min);*

*};*

*/\* Prekondisi : Tabel T tidak kosong \*/*

*/\* Mengirimkan nilai minimum tabel \*/*

*/\* \*\*\* Mengirimkan indeks elemen bernilai ekstrem \*\*\* \*/*

*IdxType IdxMaxTab (TabInt T){*

*int i;*

*for (i = (GetFirstIdx(T)+1); i <= (GetLastIdx(T)); i++){*

*if (T.TI[i] == ValMax(T)){*

*return(i);*

*}*

*}*

*};*

*/\* Prekondisi : Tabel T tidak kosong \*/*

*/\* Mengirimkan indeks i dengan elemen ke-i adalah nilai maksimum pada tabel \*/*

*IdxType IdxMinTab (TabInt T){*

*int i;*

*for (i = (GetFirstIdx(T)+1); i <= (GetLastIdx(T)); i++){*

*if (T.TI[i] == ValMin(T)){*

*return(i);*

*}*

*}*

*};*

*/\* Prekondisi : Tabel tidak kosong \*/*

*/\* Mengirimkan indeks i \*/*

*/\* dengan elemen ke-i nilai minimum pada tabel \*/*