*#include<stdio.h>*

*#include "arrayMhs.h"*

*#include <stdlib.h>*

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

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

*void MakeEmpty (TabMhs \*T){*

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

*}*

*int NbElmt (TabMhs T){*

*return T.Neff;*

*}*

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// return T.Neff;*

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

*int MaxNbEl (TabMhs T){*

*return IdxMax - IdxMin + 1;*

*}*

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// return IdxMax - IdxMin + 1;*

*IdxType GetFirstIdx (TabMhs T){*

*return IdxMin;*

*}*

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// return IdxMin;*

*IdxType GetLastIdx (TabMhs T){*

*return T.Neff;*

*}*

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

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// return T.Neff;*

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

*return T.TI[i];*

*}*

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// return T.TI[i];*

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

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

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

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

*for(int i = IdxMin; i <= Tin.Neff; i++){*

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

*}*

*}*

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

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

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

*// for(int i = IdxMin; i <= Tin.Neff; i++){*

*//     (\*Tout).TI[i] = Tin.TI[i];*

*// }*

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

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

*if ((\*T).Neff < i){*

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

*}*

*}*

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

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

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

*// if ((\*T).Neff < i){*

*//     (\*T).Neff = i;*

*// }*

*void SetNeff (TabMhs \*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 \*/*

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

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

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

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

*if ((i >= IdxMin) && (i <= IdxMax)){*

*return true;*

*}*

*else {*

*return false;*

*}*

*}*

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

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// if ((i >= IdxMin) && (i <= IdxMax)){*

*//     return true;*

*// } else {*

*//     return false;*

*// }*

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

*if ((i >= IdxMin) && (i <= T.Neff)){*

*return true;*

*} else {*

*return false;*

*}*

*}*

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

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

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// if ((i >= IdxMin) && (i <= T.Neff)){*

*//     return true;*

*// } else {*

*//     return false;*

*// }*

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

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

*boolean IsEmpty (TabMhs T){*

*if (T.Neff > 0){*

*return false;*

*} else {*

*return true;*

*}*

*}*

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// if (T.Neff > 0){*

*//     return false;*

*// } else {*

*//     return true;*

*// }*

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

*boolean IsFull (TabMhs T){*

*if (T.Neff > 0){*

*return false;*

*} else {*

*return true;*

*}*

*}*

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

*/\* Untuk membantu pengerjaan praktikum, silahkan gunakan kode berikut untuk fungsi ini \*/*

*// if (T.Neff == IdxMax){*

*//     return true;*

*// } else {*

*//     return false;*

*// }*

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

*void TulisIsi (TabMhs T){*

*if(IsEmpty(T)){*

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

*}*

*else{*

*int i;*

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

*printf("%s | %s | %.2f\n",T.TI[i].nama,T.TI[i].nim,T.TI[i].nilai);*

*}*

*}*

*}*

*/\* 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 \*/*

*/\* Contoh print tabel*

*Test | 12321123 | 3.61*

*Ini Juga Test | 12321124 | 3.21*

*Test Lagi | 12321125 | 3.11*

*\*/*

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

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

*ElType MakeMahasiswa (char\* Nama, char\* NIM, float Nilai){*

*ElType M;*

*M.nama = Nama;*

*M.nim = NIM;*

*M.nilai = Nilai;*

*return M;*

*}*

*/\* Membentuk sebuah Mahasiswa dari komponen-komponennya \*/*

*/\* I.S. Nama, NIM, Nilai terdefinisi \*/*

*/\* F.S. Mahasiswa M terbentuk dengan Nama, NIM, Nilai yang sesuai \*/*

*/\* \*\*\*\*\*\*\*\*\*\* OPERATOR STATISTIK MAHASISWA \*\*\*\*\*\*\*\*\*\* \*/*

*float RataRata (TabMhs T){*

*float sum=0;*

*int i;*

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

*sum+=T.TI[i].nilai;*

*}*

*return(sum/NbElmt(T));*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan nilai rata-rata dari elemen tabel \*/*

*float Max (TabMhs T){*

*float max = T.TI[1].nilai;*

*int i;*

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

*if (T.TI[i].nilai > max){*

*max = T.TI[i].nilai;*

*}*

*}*

*return max;*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan nilai maksimum dari elemen tabel \*/*

*float Min (TabMhs T){*

*float min = T.TI[1].nilai;*

*int i;*

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

*if (T.TI[i].nilai < min){*

*min = T.TI[i].nilai;*

*}*

*}*

*return min;*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan nilai minimum dari elemen tabel \*/*

*char \*MaxNama (TabMhs T){*

*int i,count=0;*

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

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

*count++;*

*}*

*}*

*if(count == 1){*

*return(T.TI[i].nama);*

*}*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan nama mahasiswa dengan nilai tertinggi \*/*

*/\* Jika ada lebih dari satu mahasiswa dengan nilai tertinggi, maka yang diambil yang nim lebih rendah \*/*

*char \*MinNama (TabMhs T){*

*int i,count=0;*

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

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

*count++;*

*}*

*}*

*if(count == 1){*

*return(T.TI[i].nama);*

*}*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan nama mahasiswa dengan nilai terendah \*/*

*/\* Jika ada lebih dari satu mahasiswa dengan nilai terendah, maka yang diambil yang nim lebih rendah \*/*

*int JumlahJurusan (TabMhs T, char\* Jurusan){*

*int i,j,count,check;*

*char nim\_titip[3];*

*count = 0;*

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

*{*

*for (j = 0; j < 3;j++)*

*{*

*nim\_titip[j] = T.TI[i].nim[j];*

*}*

*check = strcmp(nim\_titip, \*Jurusan);*

*if (check == 0)*

*{*

*count++;*

*}*

*}*

*return count;*

*}*

*/\* I.S. Tabel T tidak kosong \*/*

*/\* F.S. Menghasilkan jumlah mahasiswa yang berasal dari jurusan Jurusan, yang berbentuk 3 angka (contoh: 182) \*/*

*/\* Jika tidak ada mahasiswa yang berasal dari jurusan Jurusan, maka menghasilkan 0 \*/*