

There is data of a student with their GPA history until this semester. If there is 0 GPA inside the array of IPK, it indicates that the student took a leave of absence in that semester. Students are only given the opportunity to take one time off from college.

Create an ADT student that has the following procedure functions:

1. ADD NEW IPK
2. SHOW IPK
3. FIND LEAVE OF ABSENCE

ANSWER

1. **STUDENT.h**

#ifndef SLL\_H\_INCLUDED

#define SLL\_H\_INCLUDED

#include<iostream>

**Type student : < Name : string,**

**IPK : float,**

**N, NMax : int >**

**Procedure add (input/output s : student, input newGpa: float)**

*{i.s there is data of 1 student and a new GPA. The Array of IPK maybe full  
F.S the new gpa will be added inside student data}*

**Procedure show (input s : student)**

*{I.s there is data of 1 student  
F.S all the ipk of the student will be appear in the monitor except 0 GPA}*

**Function find (s : student) int**

*{the function will return the index of 0 GPA. If there is no 9 GPA, the function will return -1}*

1. **STUDENT.cpp**

**#include student.h**

**Procedure add (input/output s : student, input newGpa : float)**

*{i.s there is data of 1 student and a new GPA. The Array of IPK maybe full  
F.S the new gpa will be added inside student data}*

**Dictionary**

**Algorithm**

If ( S.N <= NMAX ) then *//check if the data is full*

S.N S.N + 1

S.IPK  newGPA

Else

Output(“FULL”)

**Procedure show (input s: student)**

*{I.s there is data of 1 student  
F.S all the ipk of the student will be appear in the monitor except 0 GPA }*

**Dictionary**

i : 0

**Algorithm**

for i < S.N to i  i + 1 do

if (S.IPK != 0 ) then

output (S.IPK)

**Function find (s : student)** **integer**

*{the function will return the index of 0 GPA. If there is no 9 GPA, the function will return -1}*

**Dictionary**

//none

**Algorithm**

i=1

while (i < S.NMAX + 1) AND (S.IPK != 0) DO

i++

if (S.N == 9 && S.IPK == 0) then

return i

else

return -1

1. **MAIN.cpp**

**#include student.h**

**Dictionary**

S : student

newGpa : float

**Algorithm**

//set the student’s name (from user)

Input(student)

//set the N

Input(S.N)

//set the NMAX

Input(S.NMAX)

//get first newGPA from user

Input(newGpa)

//Call the add procedure

add(S, newGpa)

//get second newGPA from user

Input(newGPA)

//Call the add procedure

add(S, newGPA)

//get third newGPA from user

Input(newGPA)

//Call the add procedure

add(S, newGPA)

//call procedure show

show(S)

//call find function

//dibawah

//show to user what semester the student took some time off

Output(find(s))