Program 2

This program calculates a student’s Score for Academic Record.

Input to this program:

* Rounded Weighted Average GPA ( output from Program 1 )
* RightEnd ( 1-dim. Array); the index of the RightEnd array ranges from

15 to 30; the values stored in the RightEnd array range from 3.51 to 4.0.

RightEnd(30)=4.0

RightEnd(29)=3.98

RightEnd(28)=3.94

RightEnd(27)=3.91

RightEnd(26)=3.88

RightEnd(25)=3.84

RightEnd(24)=3.81

RightEnd(23)=3.78

RightEnd(22)=3.74

RightEnd(21)=3.71

RightEnd(20)=3.68

RightEnd(19)=3.64

RightEnd(18)=3.61

RightEnd(17)=3.58

RightEnd(16)=3.54

RightEnd(15)=3.51

This Program prints as output :

* Rounded Weighted Average GPA: ( input value )
* Score for Academic Record : ( output value )

Use as input the output values calculated from the 5 Test Data cases in Program 1.

Algorithm for Program 2

This algorithm uses the student’s Rounded Weighted Average GPA to search in the RightEnd array and find the Score for Academic Record ( the index of the array ).

For example, if the student’s Rounded Weighted Average GPA is 3.53, then the Score for Academic record is 16, which is the index of RightEnd (16 )=3.54 ; the Rounded Weighted Average GPA of 3.53 is less than or equal to 3.54 the value stored in the RightEnd array at index 16.

The search in the RightEnd array can be implemented as a while loop ; it finds the corresponding score for a student’s Rounded Weighted Average GPA.

Let i be the index of the RightEnd array that ranges from 15 to 30.

i= 15 ( start search at index 15 )

Let Flag be a Boolean variable initialized to true.

While ( i <= 30 and Flag ) do

{ if ( Rounded Weighted Average GPA <= RightEnd ( i )

{ Print “ Score for Academic Record : ” i ;

Flag= false; }

i = i + 1 ; }