

Data Structure Homework

GWid: G45472436

Name: Shiyun Liu

(*Note: Click the application file in the “\bin\Debug\” folder to run every code.)

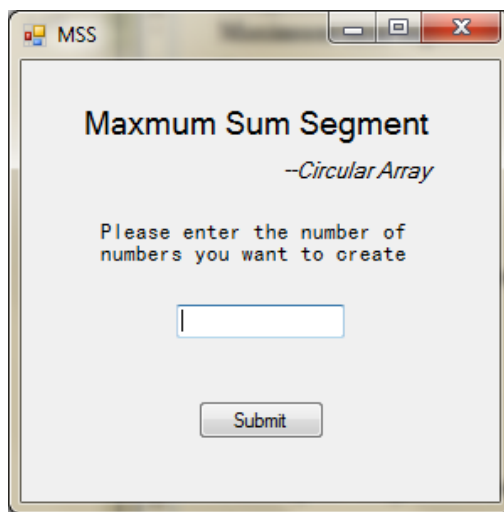
Assignment #1

Maximum sum segment

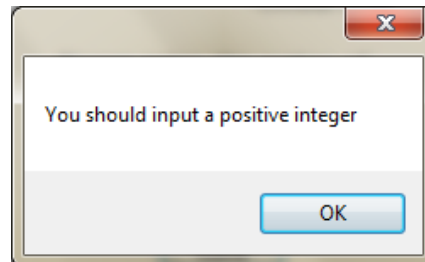
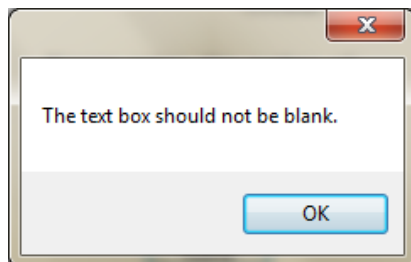
Answer:

Please see the code application.

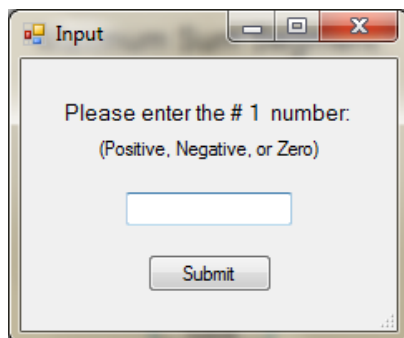
(1) Enter the number of numbers:



You should input a positive integer, blank or any other characters will cause a message box windows to give you a hint.

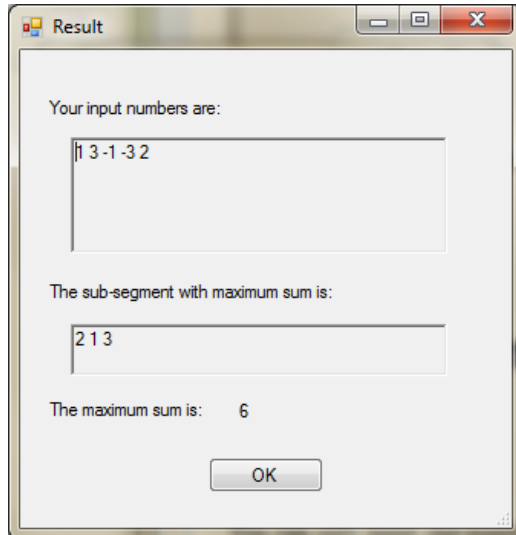


If we input “5”, another window will occur to make you input the numbers.



The input text box also has a check method to make you input an available number. You can only input one number at one time. In this case, the input window will occur 5 times to get the input numbers. And we input “1”, “3”, “-1”, “-3”, “2”.

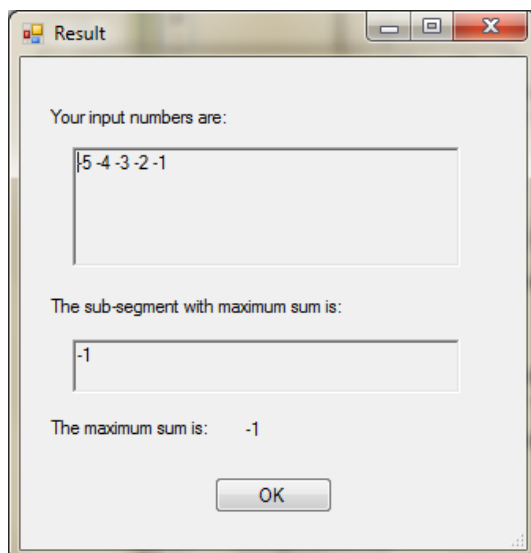
After finished, you will get a result:



(2) Special cases:

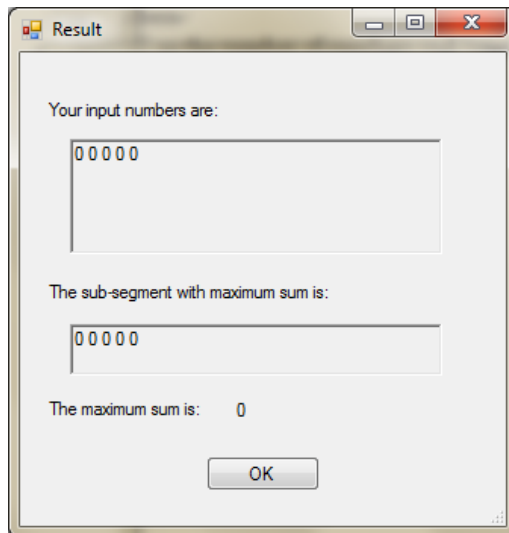
<1> All negative numbers

We also enter “5” as the number of numbers and input “-1”, “-2”, “-3”, “-4”, “-5”. Then the result will be:



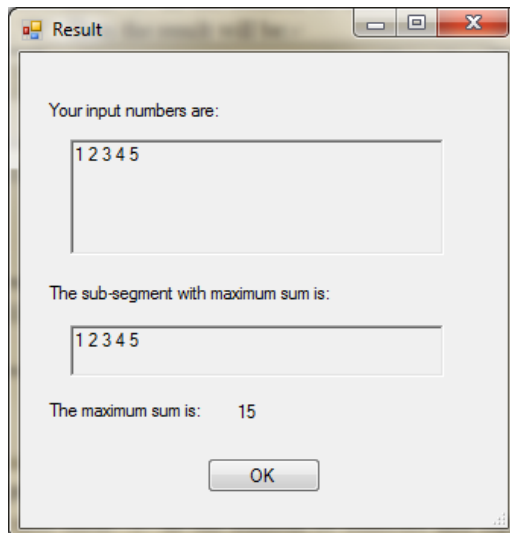
<2> All zeros

We also enter “5” as the number of numbers and input “0”, “0”, “0”, “0”, “0”. Then the result will be:



<3> All positive number:

We also enter "5" as the number of numbers and input "1", "2", "3", "4", "5". Then the result will be:



Assignment #2

Finding a sink

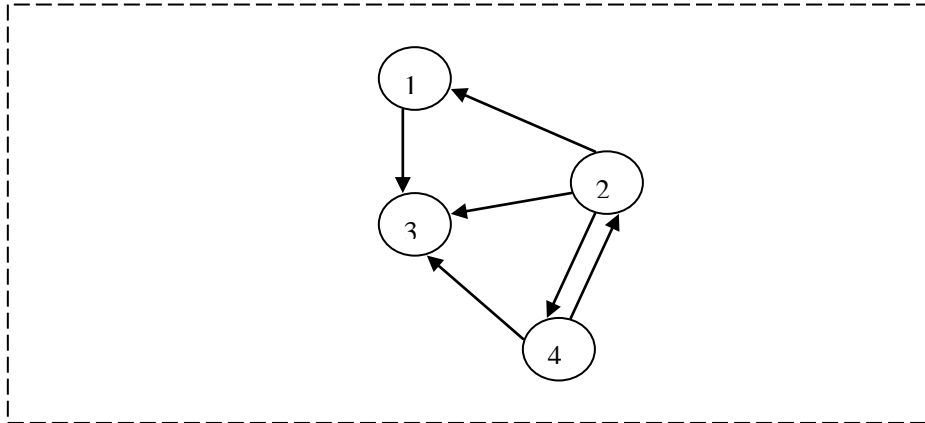
Answer:

Please see the code application. (In the class, professor said that "sink" means "universal sink.")

(1) First, you need to input the number of vertices of a directed graph.

(The input text box also has a check method to make you input an available number.)

Assume the directed graph is:



There are 4 vertices.

Sink

Find a sink

Please enter the number of vertices of a directed graph:

4

Submit

(2) Input the adjacency matrix.

(The input rich text box also has a check method to make you input available numbers.)

In this case, the adjacency matrix is:

0 0 1 0

1 0 1 1

0 0 0 0

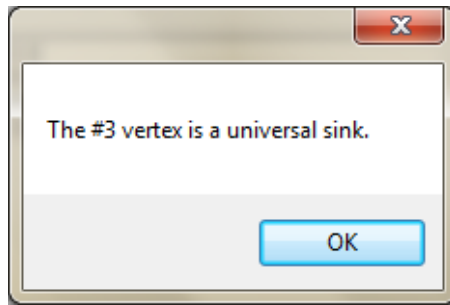
0 1 1 0

Adjacency Matrix

Please enter the adjacency matrix of the 4x4 Directed Diagram.
(Note that use a space " " between each element)

0 0 1 0
1 0 1 1
0 0 0 0
0 1 1 0

Submit



And if there is no universal sink, like

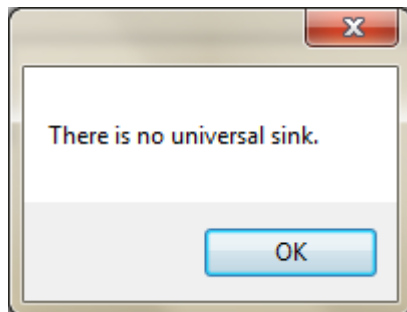
0 0 1 0

1 0 0 1

0 0 0 0

0 1 1 0

The result will be



Assignment #3

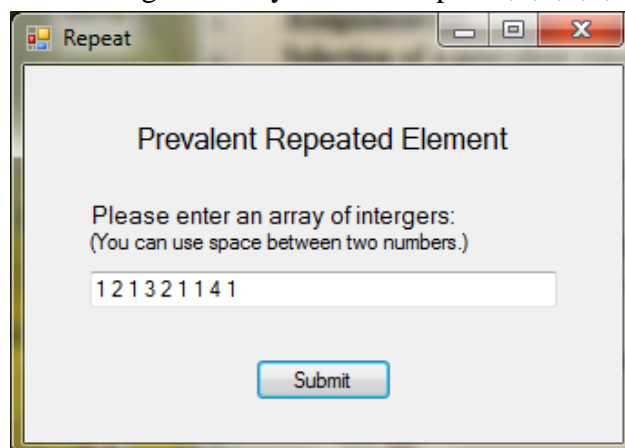
Selection of a prevalent repeated element

Answer:

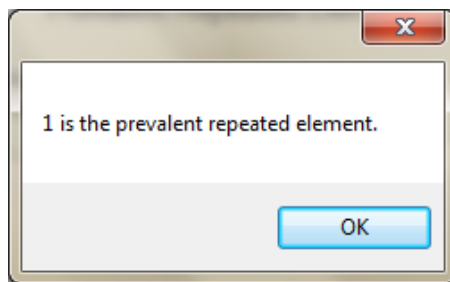
Please see the code application.

- (1) Enter the array of N integer numbers with space between each other.
(The input text box has a check method to make you input available numbers in a correct format.)

We use the given array as an example: 1,2,1,3,2,1,1,4,1.



The result will be:

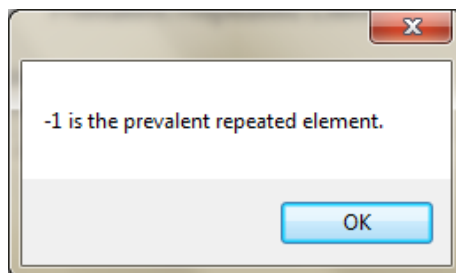


(2) Special cases:

<1> If the array contains negative numbers, like

-1,2,-1,3,2,-1,-1,4,1

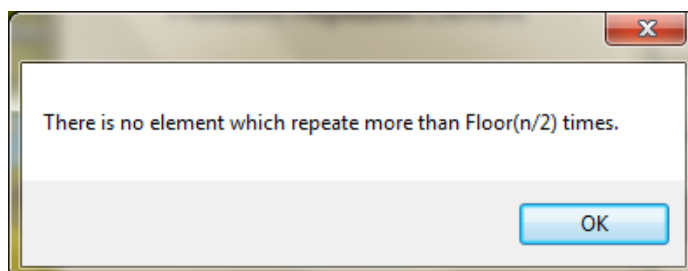
The result is:



<2> If there is no number which repeat more than $\lfloor N/2 \rfloor$ times, like

1,-2,-1,3,2,4,5,4,6

The result is:



Assignment #4

Virtual Casino

Answer:

Please see the code application.

(1) This problem defines that a deck of cards has 52 cards.



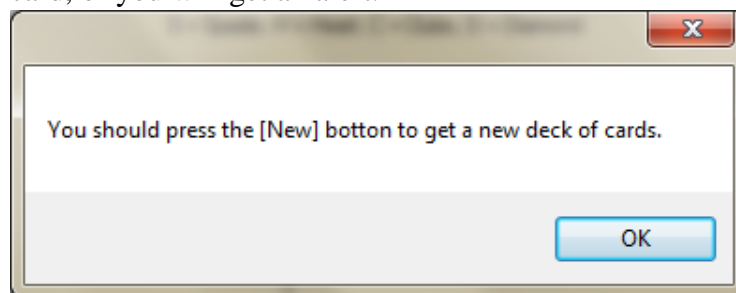
There are three buttons:

New --- Get a new deck of 52 cards

Shuffle --- Shuffle the current cards

Serve --- Give cards to a given number of players

Before pressing "Shuffle" and "Serve", you need to press "New" to get a new card, or you will get an alert.



(2) Press "New" button

The panel on the right side will display a new deck of cards sequentially.



(3) Press "Shuffle" Button

The current cards will be shuffled and the panel will display the shuffled cards and you can shuffle any times you want.



(4) Press "Serve" Button

If you press "Serve" Button a window will occur to make you decide the service method.

(The input text box has a check method to make you input available numbers.)

Servation Form

Please input the number of players:

Please select the servation form:

☐ Use all of the cards

☐ Give each person the same number of cards
(Use as most cards as you can)

☐ Give each person the same number of cards
(You can set the specified number by yourself)

Submit

You can input the number of players, like 5.

And there are three service forms:

<1>“User all of the cards” means give all cards to the players with no remain.

The number of cards of each player may not be the same.

For example, give 52 cards to 5 players, player1 and player2 will have 11 cards and others will have 10 cards.



<2> “Give same card and use as most as you can” means you need to give players as many cards as you can and the plays will have the same number of cards.

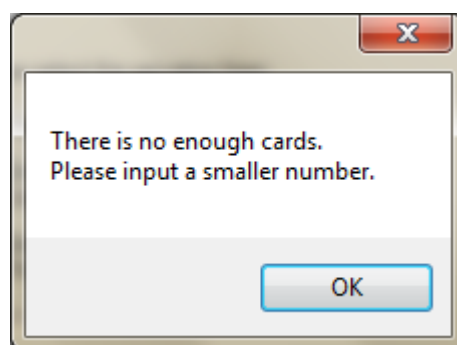
For example, give 52 cards to 5 players, all players will have 10 cards.



- <3> “Set the specified number” means you can give any number of cards to the players and each person will have the same number of cards. For example, give 3 cards to each of the 5 players, all players will have 3 cards.



If you input a large number so that it needs more cards than 52, you will get an alert.



Assignment #5

Finding the length of a non-repeating interval for a pseudo-random generator

Answer:

For the pseudo-random numbers, some of them will occur repeatedly. We can imagine it as that there is a linked list with a recycle.

In order to find the length of a non-repeating interval, we need to find the enter point of the recycle.

We can use two pointers: SLOW and FAST. Both of them start at the head point. The SLOW pointer runs one step each time, while FAST pointer runs two steps each time. If there is surely a cycle in the linked list, the two pointers will meet at some point in this cycle and the Fast pointer have traversed the cycle many times.

Assume that “X” denotes the distance between head and enter point, “L” denotes the distance from the enter point to the meet point, “C” denotes the distance of the cycle, and “n” denotes the traversed times of the cycle for the Fast pointer.

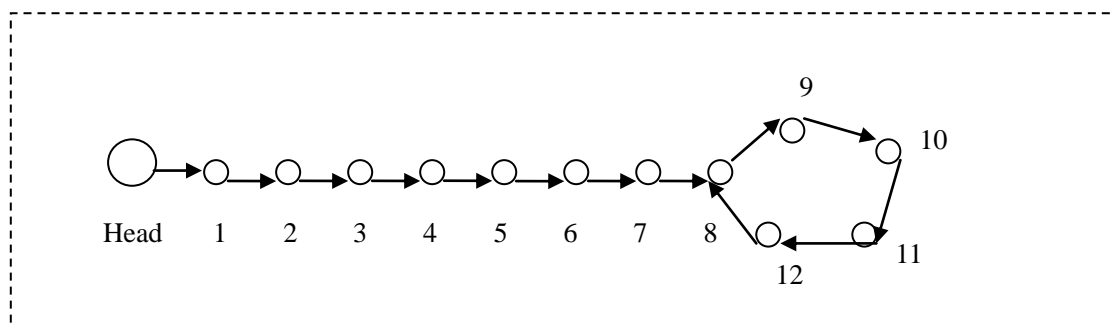
We can find that $2*(X + L) = n*C + (X+L)$.

So $X = n*C - L = (n-1)*C + (C-L)$ and $(C-L)$ is the distance from meet point to the enter point.

So if we put another two pointers PS and PM, they all go one step each time and PS starts from head and PM starts from the meet point. The two pointers will meet at the enter point of cycle.

For example, here is a linked list with a cycle like the figure below.

traverse



SLOW	1	2	3	4	5	6	7	8	9	10
FAST	2	4	6	8	10	12	9	11	8	10

They meet at the point 10.

PS	1	2	3	4	5	6	7	8
PM	11	12	8	9	10	11	12	8

They meet at the point 8.

So point 8 is the enter point of the cycle and the length of the non-repeating interval of the lined list is 8.