學號：110060007
姓名：黃俊穎

EE2410 Data Structure Hw #1 (Chapter 1~4 of textbook)

due date 4/30/2023

Format: Use a text editor to type your answers to the homework problem. You need to submit your HW in a DOCX file named as **Hw1-SNo.docx or Hw1-SNo.pdf**, where SNo is your student number. Submit the **Hw1-SNo.docx or Hw1-SNo.pdf** file via eLearn. Inside the file, you need to put the **header and your student number, name (e.g., EE2410 Data Structure Hw #1 (Chapter 1~4 of textbook) due date 4/30/2023 by SNo, name)** first, and then the **problem** itself followed by your **answer** to that problem, one by one. The grading will be based on the correctness of your answers to the problems, and the **format**. Fail to comply with the aforementioned format (file name, header, problem, answer, problem, answer,…), will certainly degrade your score. If you have any questions, please feel free to ask.

1. (2%) Determine the frequency counts for all statements (by step table) in the following two program segments:

   Code (a):

```
for(i=1; i<=n; i++)
    for(j=1; j<=i; j++)
        for(k=1; k<=j; k++)
            x++;
```

Answer:

Code (a)	s/e	freq	subtotal
for (i=1; i<=n; i++)	1	$n+1$	$n+1$
for (j=1; j<=i; j++)	1	$\sum_{k}^{n} k+1$	$\frac{n^2+3n}{2}$
for(k=1; k<=j; k++)	1	$\sum_{l=1}^{n}\sum_{k=1}^{l} k+1$	$\frac{n^3+6n^2+5n}{6}$
x++;	1	$\sum_{l=1}^{n}\sum_{k=1}^{l} k$	$\frac{n^3+3n^2+2n}{6}$
Total	$\frac{n^3+6n^2+11n+3}{3}$		

學號：110060007
姓名：黃俊穎

Code (b)

```
i=1;
while(i<=n)
{
    x++;
    i++;
}
```

Answer:

Code (a)	s/e	freq	subtotal
i=1	1	1	1
while(i<=n)	1	$n+1$	$n+1$
x++;	1	n	n
i++;	1	n	n
Total	$3n+2$		

2. (2%) For the function Multiply() shown below,
   (a) Obtain the step count for the function using the step table method.
   (b) Repeat (a) but use Θ() notation instead of exact step counts and frequency counts..

```
void Multiply(int **a, int **b, int **c, int m, int n, int p)
{
    for(int i=0;i<m;i++)
        for(int j=0; j<p; j++)
        {
            c[i][j] = 0;
            for(int k=0;k<n;k++)
              c[i][j] += a[i][k] * b[k][j];
        }
}
```

Answer:

2.(a)	s/e	freq	subtotal
for(int i=0;i<m;i++)	1	$m+1$	$m+1$
for(int j=0; j<p; j++)	1	$m(p+1)$	$m(p+1)$
c[i][j] = 0;	1	mp	mp
for(int k=0;k<n;k++)	1	$mp(n+1)$	$mp(n+1)$
c[i][j] += a[i][k] * b[k][j];	1	mpn	mpn
Total	$2mpn+3mp+2m+1$		

學號：110060007
姓名：黃俊穎

2.(b)	s/e	freq	subtotal
for(int i=0;i<m;i++)	1	$\theta(m)$	$\theta(m)$
for(int j=0; j<p; j++)	1	$\theta(mp)$	$\theta(mp)$
c[i][j] = 0;	1	$\theta(mp)$	$\theta(mp)$
for(int k=0;k<n;k++)	1	$\theta(mpn)$	$\theta(mpn)$
c[i][j] += a[i][k] * b[k][j];	1	$\theta(mpn)$	$\theta(mpn)$
Total	$\theta(mpn)$		

3. (5%) For each of the complexity expression below, determine its overall complexity. For example, given expression $2n + 3n$, the overall complexity should be $O(n)$, not $O(n^2)$, i.e., use the smallest possible class of functions.

Answer:

(a) $2n^2 - 3n = O(n^2)$

(b) $n! + 2^n = O(n!)$

(c) $5n^2 + n \log n = O(n^2)$

(d) $n^{1.001} + n \log n = O(nlogn)$

(e) $5n^3 - 3n^2 \log n + 2n = O(n^3)$

4. (6%) **(Recursion and Performance Analysis)** According to the below recursive method, answer the following questions.

```
int recurse(int n)
{
    if (n <= 0)
    return 1;
    else
    return 1 + recurse(n/2);
}
```

a) What does the method return for n=8?

b) How many times will your method be called recursively for n=256?

c) What is the algorithm complexity of the recursive method in big Oh notation?

Answer:

(a) $1 + T(4)$

$= 1 + 1 + T(2)$

$= 1 + 1 + 1 + T(1)$

$= 1 + 1 + 1 + 1 + T(0)$

$= 1 + 1 + 1 + 1 + 1$

$= 5$

(b) 10. By(a), we can know that as n = 8 = 2^3, it takes 5 (3+2) methods for return. Similarly, for n = 256 = 2^8, it should take 10 (8+2) methods for return.
(c) $O(logn)$, since the value n declines logarithmically when doing recursive return.

5. (2%) Given an algorithm that solves a problem in three phases. The first phase takes $O(100n)$ to input the data of size n. The second phase takes $O(n \log n)$ to process the data. The third phase takes $O(\log n)$ to output the data.
   (a) What is the complexity of the algorithm?
   (b) If the data size is 10, which phase is most likely to take the longest time to execute?

   Answer:
   (a) $O(nlogn)$, as its level is the highest of these three.
   (b) The first phase: $100 * 10 = 1000$
   The second phase: $10 * log10 = 10$
   The third phase: $log10 = 1$
   Thus, the first phase takes the longest time.

6. (8%)
   (a) Given a list of objects stored in a sorted array, describe an algorithm to search as efficiently as possible for an object based on a key. (Note that the key type matches the field type by which the array is sorted.) You may describe your algorithm in English, pseudocode, and/or C++ code, as long as your description is clear.
   (b) What will be the Big O running time of the algorithm you described in (a)? Why?
   (c) Given a list of objects stored in a unsorted array, describe an algorithm to search as efficiently as possible for an object based on a key. You may describe your algorithm in English, pseudocode, and/or C++ code, as long as your description is clear.
   (d) What will be the Big O running time of the algorithm you described in (c)? Why?

學號：110060007
姓名：黃俊穎

Answer:

(a)

```
int BinarySearch(int *a, const int x, const int l)
{
    int left = 0, right = l - 1;
    while(left <= right) {
        int mid = (left + right) / 2;
        if (x < a[mid]) right = mid - 1;
        else if (x > a[mid]) left = mid + 1;
        else return mid;
    }
    return -1;
}
```

By binary search method, we call an array a with length l, then find the target x. First, we set a standard index mid (middle element of zoomed scope) of the array. Since the array is sorted, if x is larger or smaller than middle, we will focus on rest half range to keep finding target x, which is 1/2 of previous round's scope. If x is found, we will return its index position, and if not, we will return -1 for the case.

(b) Time complexity = O($logn$)

(c)

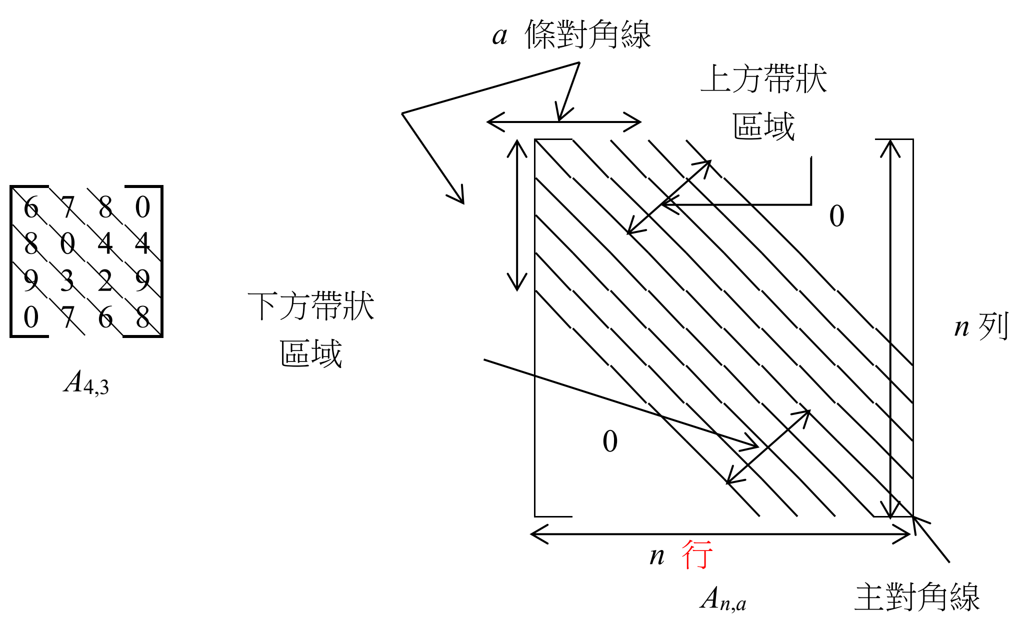
```
int LinearSearch(int *a, const int x, const int l)
{
    for (int i = 0; i < l; i++) {
        if (x == a[i]) return i;
    }
    return -1;
}
```

By linear search method, we call an array a with length l, then find the target x. We will scan through the array and find x's index position. If x is found, then return the index position, and if not, we will return -1 for the case.

(d) It will be O(n) because the worst case is that when x is at the last position of the array or x doesn't exist. then it would take the time complexity n of the traversal to obtain the output.

學號：110060007
姓名：黃俊穎

7. (6%) A square band matrix $A_{n,a}$ is an n x n matrix A in which all the nonzero terms lie in a band centered around the main diagonal. The band includes a-1 diagonals below and above the main diagonal as shown below.



(a) How many elements are there in the band of $A_{n,a}$?
(b) What is the relationship between i and j for element A_{ij} in the band of $A_{n,a}$?
(c) Assume that the band of $A_{n,a}$ is stored sequentially in an array b by diagonals starting with the lowermost diagonal. Thus $A_{4,3}$ above would have the following representation:

b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]	b[10]	b[11]	b[12]	b[13]
9	7	8	3	6	6	0	2	8	7	4	9	8	4
A_{20}	A_{31}	A_{10}	A_{21}	A_{32}	A_{00}	A_{11}	A_{22}	A_{33}	A_{01}	A_{12}	A_{23}	A_{02}	A_{13}

Obtain an addressing formula for the location of an element A_{ij} in the lower band of $A_{n,a}$, e.g., LOC(A_{20}) = 0, LOC(A_{31}) = 1 in the example above.

Answer:

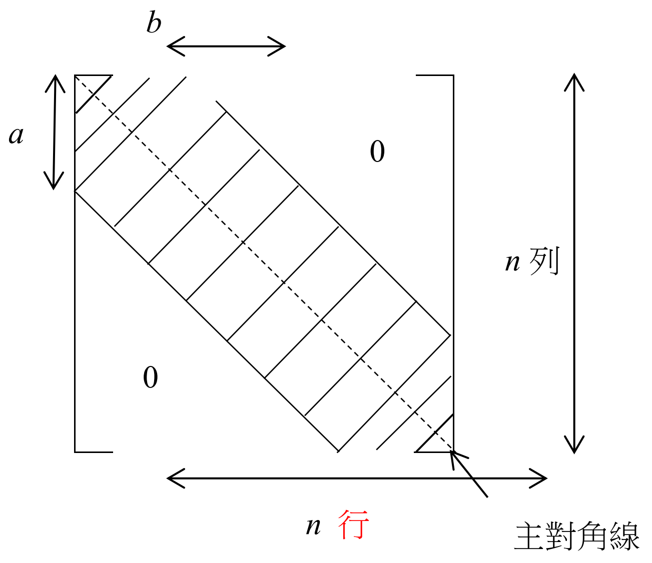
(a) $n+\sum_{i=1}^{a-1}2(n-i) = n+2n(a-1)-\frac{2a(a-1)}{2} = n(2a-1)-a(a-1)$

(b) The difference between i and j will be constant in each line. The absolute difference is the distance between the line and the diagonals.

(c) $(i-j)$ means the distance between the line the element is located to. Because there are $a-1-(i-j)$ lines below the element. Hence, elements in those lines are $\sum_{k=i-j+1}^{a-1} n-k$. Then calculate the element's position on its line, which is $j+1^{th}$. Then we can get the element is on position:

$$\left[\sum_{k=i-j+1}^{a-1} n-k\right]+j$$

8. (6%) A generalized band matrix $A_{n,a,b}$ is an n x n matrix A in which all the nonzero terms lie in a band made up of a-1 diagonals below the main diagonal, the main diagonal, and b-1 above the main diagonal as shown below.



$A_{n,\,a,\,b}$

(a) How many elements are there in the band of $A_{n,a,b}$?

(b) What is the relationship between i and j for element A_{ij} in the band of $A_{n,a,b}$?

(c) Obtain a sequential representation of the band of $A_{n,a,b}$ in one-dimensional array c.

Answer:

(a) $n+\sum_{i=1}^{a-1}(n-i)+\sum_{i=1}^{b-1}(n-i)$

$$=n+n(a-1)-\frac{a(a-1)}{2}+n(b-1)-\frac{b(b-1)}{2}$$

$$=n(a+b-1)-\frac{[a(a-1)+b(b-1)]}{2}$$

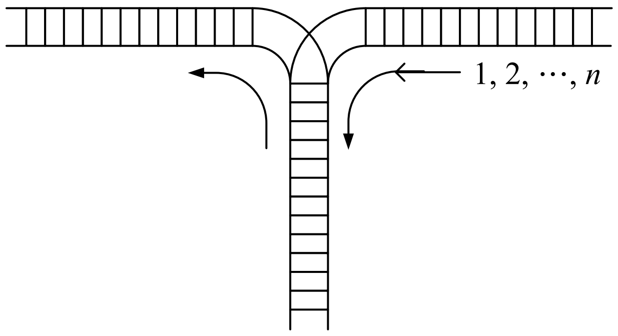
(b) The difference between i and j will be constant in each line. The absolute difference means the distance between the line and the diagonals.

(c) In the lower band and diagonals $A_{ij}=\left[\sum_{k=i-j+1}^{a-1}(n-k)\right]+j$

In the upper band $A_{ij}=\left[\sum_{k=0}^{j-i-1}(n-k)\right]+i+\sum_{k=1}^{a-1}(n-k)$

學號：110060007
姓名：黃俊穎

9. (2%) Consider the railroad switching network shown below. (textbook pp.138-139)



Railroad cars can be moved into the vertical track segment one at a time from either of the horizontal segments and then moved from the vertical segment to any one of the horizontal segments. The vertical segment operates as a **stack** as new cars enter at the top and cars depart the vertical segment from the top. Railroad cars numbered 1,2,3…,n are initially in the top right track segment. Answer the following questions for n=3 and 4 cases:

(a) What are the possible permutations of the cars that can be obtained?
(b) Are any permutations not possible? If no, simply answer no. If yes, list them all.

Answer:

(a) (n = 3) possible cases: 123, 132, 213, 231, 321
(n = 4) possible cases: 1234, 1243, 1324, 1432, 2134, 2143, 3214, 4321, 1342, 2341, 2314, 2431, 3241, 3421
(b) (n = 3) impossible cases: 312
(n = 4) impossible cases: 1423, 2413, 3124, 3142, 3412, 4123, 4132, 4213, 4231, 4312

10. (4%) A linear list of type T objects is being maintained circularly in an array with front and rear set up as for **circular queues** as described in textbook.

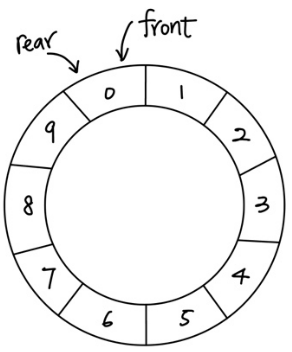
(a) Draw a diagram of the circular array showing the initial status (values of front, rear) when the linear list is created (constructed) with no elements stored yet (assume capacity = 10 is used for creating the array)
(b) Obtain a formula in terms of the array capacity, front, and rear, for the number of elements in the list.
(c) Assume the kth element in the list is to be deleted, the elements after it should be moved up one position. Give a formula describing the positions of those elements to be moved up one position in terms of k, front, rear, capacity.

Design an algorithm (pseudo code) for this Delete(int k) member function.

(d) Assume that we want to insert an element y immediately after the kth element. So the elements from the (k+1)th element on should be moved down one position in order to give space for y, which might cause insufficient capacity case in which array doubling will be needed. Describe the situation when array doubling is needed (in terms of k, front, rear, capacity). Design an algorithm (pseudo code) for this Insert(int k, T& y) member function.

Answer:

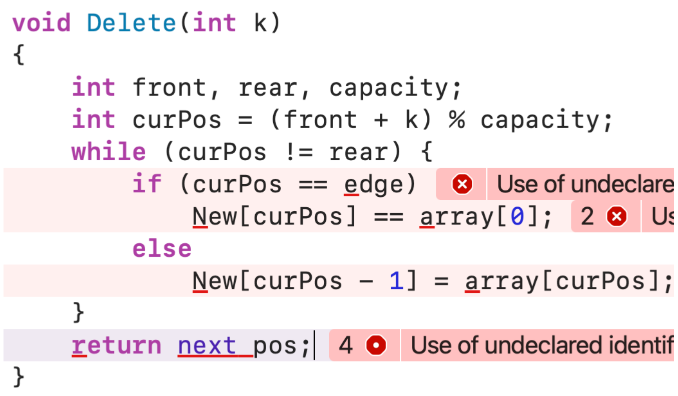
(a)



(b) Number of elements = (rear – front + capacity) % capacity

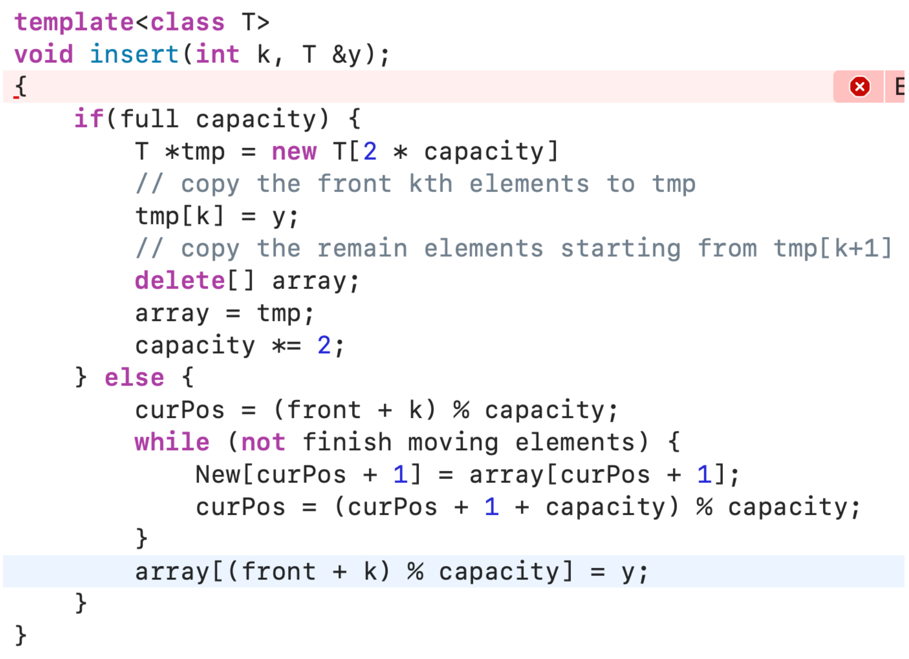
(c) The position that would be moved is NewPos = CurPos - 1, which means that

New(front + k + 1) = (front + k) . Pseudo code:



```
void Delete(int k)
{
    int front, rear, capacity;
    int curPos = (front + k) % capacity;
    while (curPos != rear) {
        if (curPos == edge)   ⊗ Use of undeclare
            New[curPos] == array[0];   2 ⊗ Us
        else
            New[curPos - 1] = array[curPos];
    }
    return next_pos;|   4 ● Use of undeclared identif
}
```

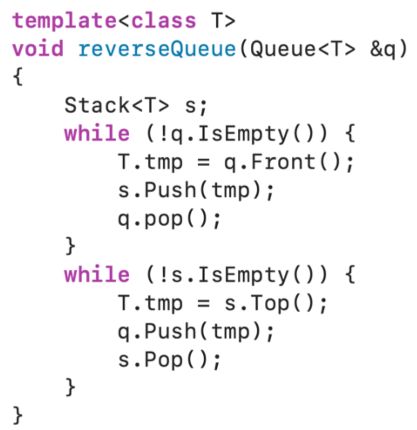
(d) The position that would be moved is NewPos = CurPos - 1, which means that New (front + k + 1) = (front + k). When doubling the size, we will copy the first kth elements to tmp and insert tmp[k] with y, then copy the rest elements. If we don’t need to couple with the size, we just need to move the elements backward with one position and insert y into the k+1th position, counting from the first element.



```
template<class T>
void insert(int k, T &y);
{
    if(full capacity) {
        T *tmp = new T[2 * capacity]
        // copy the front kth elements to tmp
        tmp[k] = y;
        // copy the remain elements starting from tmp[k+1]
        delete[] array;
        array = tmp;
        capacity *= 2;
    } else {
        curPos = (front + k) % capacity;
        while (not finish moving elements) {
            New[curPos + 1] = array[curPos + 1];
            curPos = (curPos + 1 + capacity) % capacity;
        }
        array[(front + k) % capacity] = y;
    }
}
```

11. (2%) Design an algorithm, reverseQueue, that takes as a parameter a queue object and uses a stack object to reverse the elements of the queue. The operations on queue and stack should strictly follow the ADT 3.2 Queue ADT and ADT 3.1 Stack ADT.

Answer:



```
template<class T>
void reverseQueue(Queue<T> &q)
{
    Stack<T> s;
    while (!q.IsEmpty()) {
        T.tmp = q.Front();
        s.Push(tmp);
        q.pop();
    }
    while (!s.IsEmpty()) {
        T.tmp = s.Top();
        q.Push(tmp);
        s.Pop();
    }
}
```

12. (2%) Given an integer k and a queue of integers, how do you reverse the order of the first k elements of the queue, leaving the other elements in the same relative order? For example, if k=4 and queue has the elements [10, 20, 30, 40, 50, 60, 70, 80, 90]; the output should be [40, 30, 20, 10, 50, 60, 70, 80, 90]. Design your algorithm for this task.

Answer:

```
template<class T>
void reverse_K_Queue(Queue<T> &q, int k)
{
    Stack<T> s;
    int cnt = 0;
    while (cnt < k) {
        T.tmp = q.Front();
        s.Push(tmp);
        q.pop();
        cnt++;
    }
    cnt = 0;
    while (cnt < k) {
        T.tmp = s.Top();
        q.Push(tmp);
        s.Pop();
        cnt++;
    }
}
```

13. (2%) Design a method that will take two sorted stacks A and B (min on top) and create one stack that is sorted (min on top) by merging these two stacks. You are allowed to use only the stack operations such as pop, push, size and top. No other data structure such as arrays are not allowed. You are allowed to use stacks. Assume that elements on the stack can be compared using > operator.

Answer:

```
template<class T>
Queue<T> Queue<T>::Merge(Queue<T> s2)
{
    int newSize = Size() + s2.size() + 1;
    Queue<T> tmp(newSize);
    while (Size() && s2.Size()) {
        if (Top() > s2.Top()) {
            tmp.Push(s2.Pop());
            s2.Pop();
        } else {
            tmp.Push(Top());
            Pop();
        }
    }
    while (Size()) {
        tmp.Push(Top());
        Pop();
    }
    while (s2.size()) {
        tmp.Push(s2.Top());
        s2.Pop();
    }
    return tmp;
}
```

學號：110060007
姓名：黃俊穎

14. (4%) Suppose that you are a financier and purchase 100 shares of stock in Company *X* in each of January, April, and September and sell 100 shares in each of June and November. The prices per share in these months were

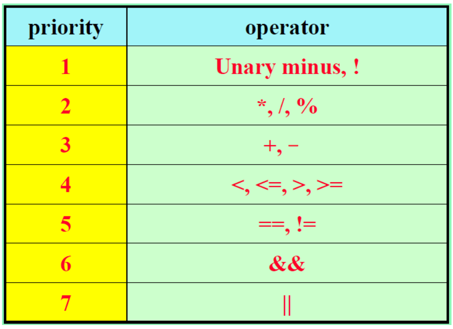
Jan	Apr	Jun	Sep	Nov
\$10	\$30	\$20	\$50	\$30

Determine the total amount of your capital gain or loss using (a) FIFO (first-in first-out) accounting and (b) LIFO (last-in, first-out) accounting [that is, assuming that you keep your stock certificates in (a) a queue or (b) a stack.]
The 100 shares you still own at the end of the year do not enter the calculation.

Answer:d

(a) $[-(10+30)+(20+30)]*100=1000$. Earn 1000 dollars.
(b) $[-(50+30)+(20+30)]*100=-3000$. Lost 3000 dollars.

15. (4%) Using the operator priorities of Figure 3.15 (shown below**)** together with those for ‘(‘ and ‘#’ to answer the following:



priority	operator
1	Unary minus, !
2	*, /, %
3	+, −
4	<, <=, >, >=
5	==, !=
6	&&
7	||

(a) In function Postfix (Program 3.19, pptx **Infix to Postfix Algorithm**), what is the maximum number of elements that can be on the stack at any time if the input expression has n operators and delimiters?
(b) What is the answer to (a) if the input expression e has n operators and the depth of nesting of parentheses is at most 6?

Answer:

(a)	If n % 9 <= 7	If n % 9 = 8
	$\frac{8n}{9}+n\,\%\,9$	$\frac{n}{9}+7$
(b)	If n <= 49	If n > 49
	$n+6$	55

學號：110060007
姓名：黃俊穎

16. (10%) Write the postfix form and prefix form of the following infix expressions:
(a) –A + B – C + D*A/B
(b) A * -B + C/D – A*C
(c) (A + B) /D + E / (F + A * D) + C
(d) A && B || C || !(E > F)
(e) !(A && !((B < C) || (C > E))) || (C < D)

Answer:

	Postfix	Prefix
(a)	A – B + C – D A * B / +	+ - + - A B C / * D A B
(b)	A B - * C D / + A C * -	- + * A - B / C D * A C
(c)	A B + D / E F A D * + / + C +	+ + * + A B D / E + F * A D C
(d)	A B && C || E F > ! ||	|| || && A B C ! > E F
(e)	A B C < C E > || ! && ! C D < ||	|| ! && A ! || < B C > C E < C D

17. (3%) Evaluate the following postfix expressions:
(a) 8 2 + 3 * 16 4 / - =
(b) 12 25 5 1 / / * 8 7 + - =
(c) 5 4 3 2 ^ + * 4 * 6 – =

Answer:

	Equation (Infix)	Answer
(a)	(8 + 2) * 3 – 16 / 4	26
(b)	12 * 25 / (5 / 1) – (8 + 7)	45
(c)	5 * (4 + 3 ^ 2) * 4 - 6	254

18. (14%) Given a template linked list **L** instantiated by the Chain class with a pointer **first** to the first node of the list as shown in Program 4.6 (textbook). The node is a ChainNode object consisting of a template data and link field.

```
template < class T > class Chain;   // 前向宣告

template < class T >
class ChainNode {
friend class Chain <T>;
private:
    T data;
    ChainNode<T>* link;
};

template <class T>
class Chain {
public:
    Chain( ) {first = 0;} //  建構子將 first 初始化成 0
    //  鏈的處理運算
    .
    .
private:
    ChainNode<T> * first;
}
```

Program 4.6

(a) **Formulate an algorithm** (pseudo code OK, C++ code not necessary) which will count the number of nodes in L. Explain your algorithm properly (using either text or graphs).

(b) **Formulate an algorithm** that will change the data field of **the kth node** (the first 1st node start at index 0) of L to the value given by Y. Explain your algorithm properly (using either text or graphs).

(c) **Formulate an algorithm** that will perform an insertion to the **immediate before of the kth node** in the list L. Explain your algorithm properly (using either text or graphs).

(d) **Formulate an algorithm** that will **delete every other node** of L beginning with node first (i.e., the first, 3rd, 5th,…nodes of L are deleted). Explain your algorithm properly (using either text or graphs).

(e) Formulate an algorithm divideMid that will divides the given list into two sublists of (almost) equal sizes. Suppose myList points to the list with elements 34 65 27 89 12 (in this order). The statement: myList.divideMid(subList); divides myList into two sublists: myList points to the list with the elements 34 65 27, and subList points to the sublist with the elements 89 12. Formulate a step-by-step algorithm to perform this task. Explain your algorithm properly (using either text or graphs).

(f) **Formulate an algorithm** that will **deconcatenate** (or **split**) a linked list L into two linked list. Assume the node denoted by the pointer variable split is to be the first node in the second linked list. Formulate a step-by-step algorithm to perform this task. Explain your algorithm properly (using either text or graphs).

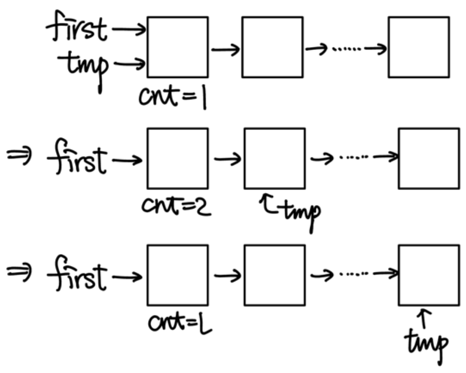
學號：110060007
姓名：黃俊穎

(g) Assume L_1 and L_2 are two chains: $L_1 = (x_1,x_2,..,x_n)$ and $L_2 = (y_1,y_2,\dots,y_m)$, respectively. **Formulate an algorithm** that can **merge** the two chains together to obtain the chain $L_3 = (x_1,y_1,x_2,y_2,\dots,x_m,y_m,x_{m+1},..,x_n)$ if n>m and $L_3 = (x_1,y_1,x_2,y_2,\dots,x_n,y_n,y_{n+1},..,y_m)$ if n<m. Explain your algorithm properly (using either text or graphs).

Answer:

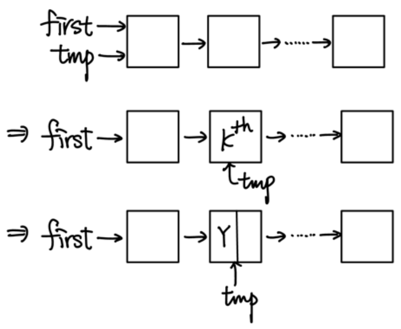
(a)

```
template<class T>
int Chain<T>::CountNodes()
{
    ChainNode<T> *tmp = first;
    int cnt = 1;
    while(tmp != last node)
    {
        cnt++;
        tmp = tmp->link; // tmp turn to the next node
    }
    return cnt; // return the number of nodes
}
```



(b)

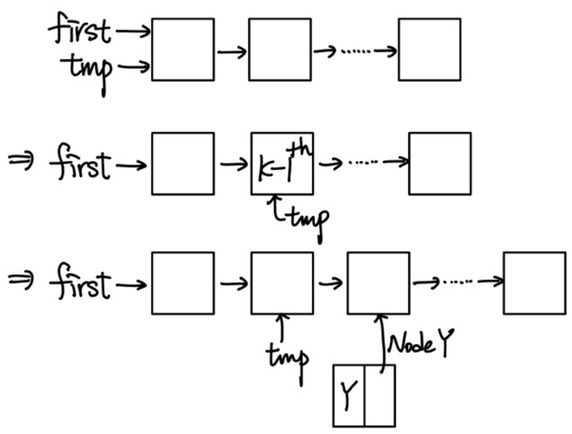
```
template<class T>
int Chain<T>::Replace(T Y, int k)
{
    ChainNode<T> *tmp = first;
    int cnt = 1;
    if (k > L) throw "out of scope";
    for (tmp != kth_node) {
        tmp = tmp->link; // tmp move to the kth node
    }
    tmp->data = Y // replace the value with Y
}
```



學號：110060007
姓名：黃俊穎

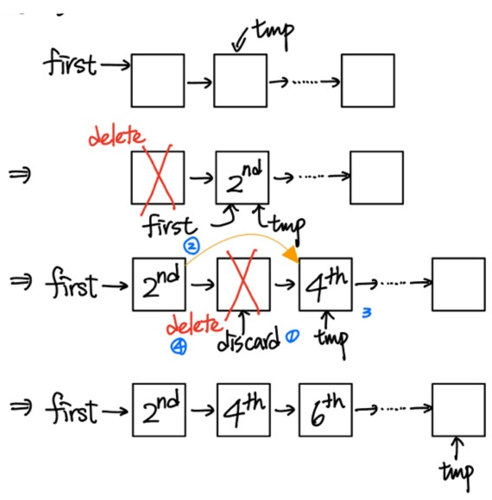
(c)

```
template<class T>
void Chain<T>::Insert(T Y, int k)
{
    ChainNode<T> *tmp = first;
    ChainNode<T> nodeY(Y,link);
    for (tmp != (k - 1)th node) {
        tmp = tmp->link; // tmp move to the (k-1)th node
    }
    nodeY->link = tmp->link;
    tmp->link - nodeY; //Insert the node Y after tmp node
}
```



(d)

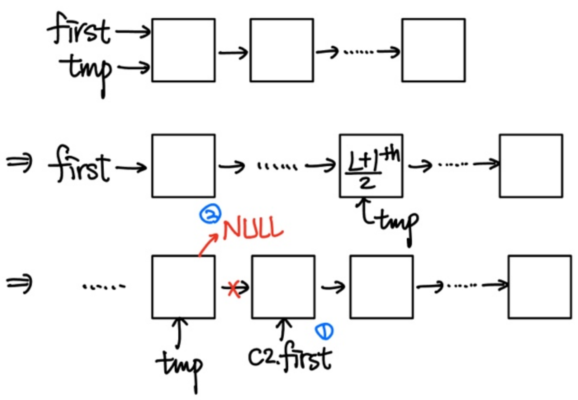
```
template<class T>
void Chain<T>::OddDelete()
{
    ChainNode<T> *tmp = first;
    delete first;
    first = tmp;
    while(tmp != last node) {
        ChainNode<T> *discard = tmp->link; // step 1
        tmp->link = tmp->link->link;       // step 2
        tmp = tmp->link;                   // step 3
        delete discard;                    // step 4
    }
}
```



學號：110060007
姓名：黃俊穎

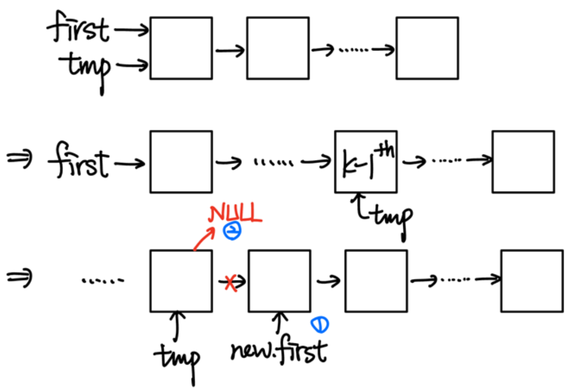
(e)

```
template<class T>
void Chain<T>::divideMid()
{
    ChainNode<T> *tmp = first;
    while (tmp != node (L+1) / 2) {
        tmp = tmp->link;
    }
    ChainNode<T> *c2;
    c2.first = tmp->link;
    tmp->link = NULL;
    return c2;
}
```

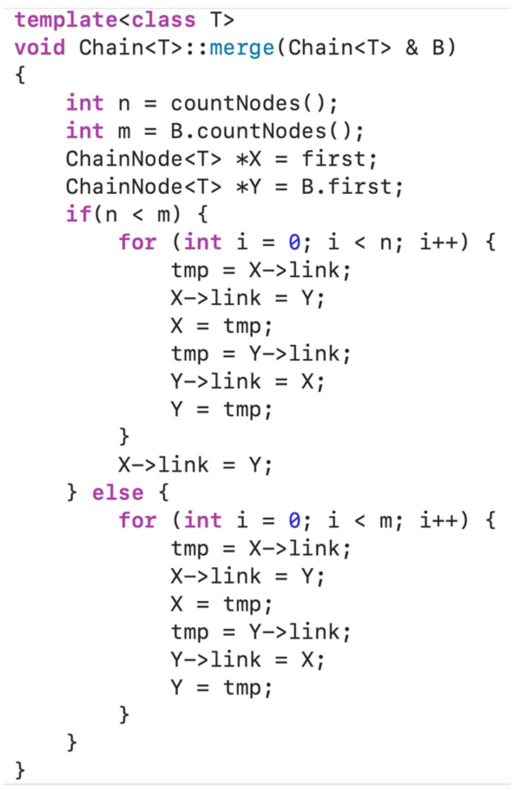


(f)

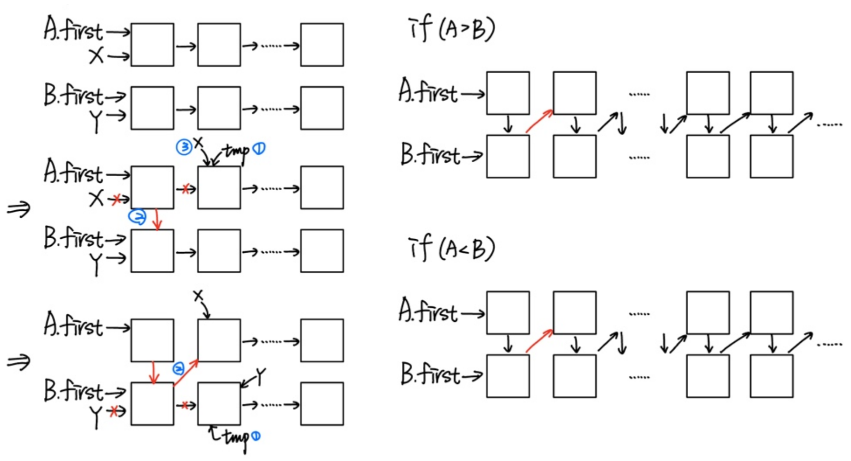
```
template<class T>
Chain<T> Chain<T>::Split(int k)
{
    Chain<T> new;
    ChainNode<T> *tmp = first;
    for (int i = 1; i < k; i++) {
        tmp = tmp->link;
    }
    new->first = tmp->link;
    tmp->link = NULL;
    return new;
}
```



(g)



```
template<class T>
void Chain<T>::merge(Chain<T> & B)
{
    int n = countNodes();
    int m = B.countNodes();
    ChainNode<T> *X = first;
    ChainNode<T> *Y = B.first;
    if(n < m) {
        for (int i = 0; i < n; i++) {
            tmp = X->link;
            X->link = Y;
            X = tmp;
            tmp = Y->link;
            Y->link = X;
            Y = tmp;
        }
        X->link = Y;
    } else {
        for (int i = 0; i < m; i++) {
            tmp = X->link;
            X->link = Y;
            X = tmp;
            tmp = Y->link;
            Y->link = X;
            Y = tmp;
        }
    }
}
```



19. (10%) Given a **circular linked list L** instantiated by class CircularList containing a private data member, **first** pointing to the first node in the circular list as shown in Figure 4.14.

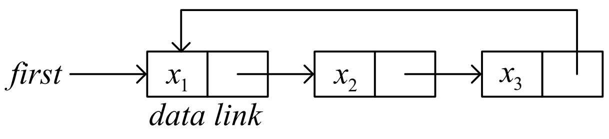
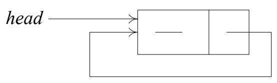


Fig. 4.14 A circular linked list

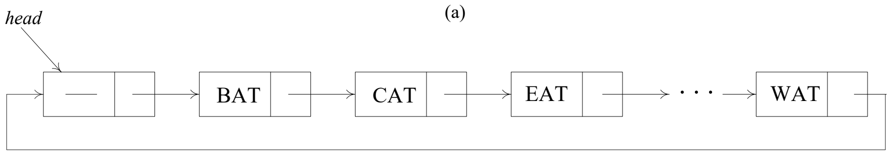
formulate algorithms (pseudo code OK, C++ code not necessary) to

學號：110060007
姓名：黃俊穎

(a) count the number of nodes in the circular list. Explain your algorithm properly (using either text or graphs)
(b) insert a new node at the front of the list. Discuss the time complexity of your algorithm. Explain your algorithm properly (using either text or graphs)
(c) insert a new node at the back (right after the last node) of the list. Discuss the time complexity of your algorithm. Explain your algorithm properly (using either text or graphs)
(d) delete the first node of the list. Discuss the time complexity of your algorithm. Explain your algorithm properly (using either text or graphs)
(e) delete the last node of the list. Discuss the time complexity of your algorithm. Explain your algorithm properly (using either text or graphs).
(f) Repeat (a) – (e) above and (b) – (g) in Problem 1 above if the circular list is modified as shown in Figure 4.16 below by introducing a dummy node, header.



(a)



(b)

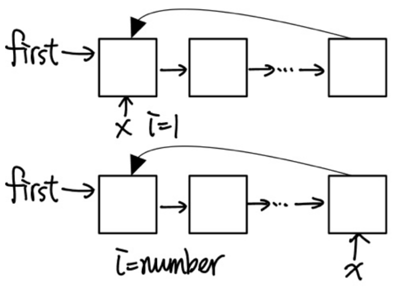
Figure 4.16

Circular list with a header node

Answer:

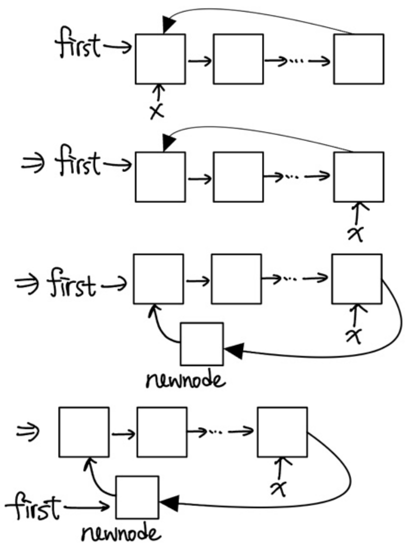
(a)

```
template<class T>
int CircularList <T>::Length() {
    ChainNode<T> *x = first;
    int i = 1;
    for (; x->link != first; i++) {
        x = x->link;
    }
    return i;
}
```



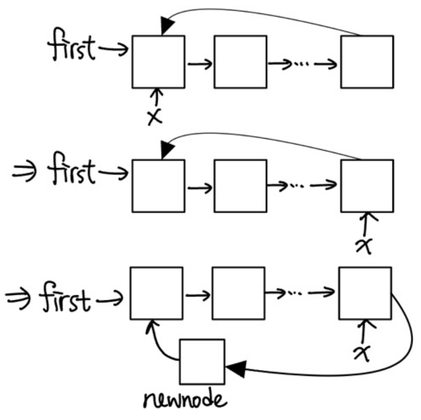
(b)

```
template<class T>
void CircularList <T>::InsertFront(ChainNode<t> new_node) {
    ChainNode<T> *x = first;
    while (x->link != first) {
        x = x->link;
    }
    x->link = new_node;
    new_node->link = first;
    first = new_node;
}
```



(c)

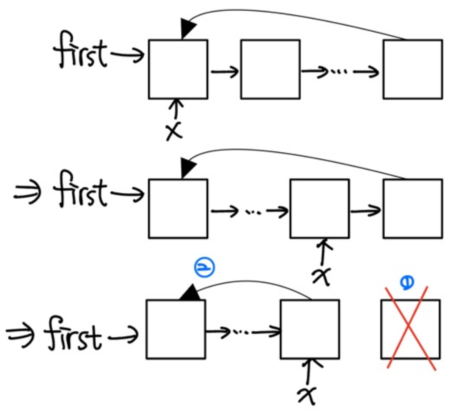
```
template<class T>
void CircularList <T>::InsertBack(ChainNode<t> new_node) {
    ChainNode<T> *x = first;
    while (x->link != first) {
        x = x->link;
    }
    x->link = new_node;
    new_node->link = first;
}
```



學號：110060007
姓名：黃俊穎

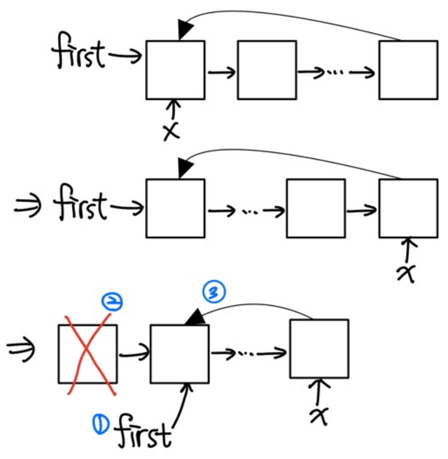
(d)

```cpp
template<class T>
void CircularList <T>::DeleteFirst() {
    ChainNode<T> *x = first;
    while (x->link != first) {
        x = x->link;
    }
    first = first->link;
    delete x->link;
    x->link = first;
}
```



(e)

```cpp
// 19(e)
template<class T>
void CircularList <T>::DeleteLast() {
    ChainNode<T> *x = first;
    while (x->link->link != first) {
        x = x->link;
    }
    delete x->link;
    x->link = first;
}
```



20. (6%) The class List<T> is shown below,

```
template <class T> class List;
template <class T>
class Node{
friend class List<T>;
private:    T data;
            Node* link;
};
template <class T>
class List{
public:
      List(){first = 0;}
      void InsertBack(const T& e);
      void Concatenate(List<T>& b);
      void Reverse();
      class Iterator{
      ….
      };
      Iterator Begin();
      Iterator End();
private:
      Node* first;
};
```

(a) Implement (pseudo code or C++) the stack data structure as a derived class of the class List<T>.

(b) Implement (pseudo code or C++) the queue data structure as a derived class of the class List<T>.

(c) Let $x_1, x_2, \ldots, x_n$ be the elements of a List<**int**> object. Each x_i is an integer. Formulate an algorithm (pseudo code OK, C++ code not necessary) to compute the expression $\sum_{i=1}^{n-5}(x_i \times x_{i+5})$

學號：110060007
姓名：黃俊穎

Answer:

(a)

```
class List<T> { // add friend class for privite members if List
    ......
    friend class Stack<T>;
    ......
}

template<class T>
Class Stack:public List<T>{
    public:
        Stack();
        void Pop();
        T& Top();
        void Push(Const T& element);
        bool IsEmpty();
    private:
    Node<T> *top;
};

template<class T>
void Stack<T>::Pop(){
    if(IsEmpty())
        throw "Stack is empty";
    else
        Node *x = top;
    top = top->link;
    delete x;
}
```

```
template<class T>
void Stack<T>::Push(const T element){
    Node *tmp = new Node;
    tmp->data = element;
    tmp->link = top;
    top = tmp;
}

template<class T>
T Stack<T>::Top(){
    return top->data;
}

template<class T>
bool Stack<T>::IsEmpty(){
    return top == 0;
}
```

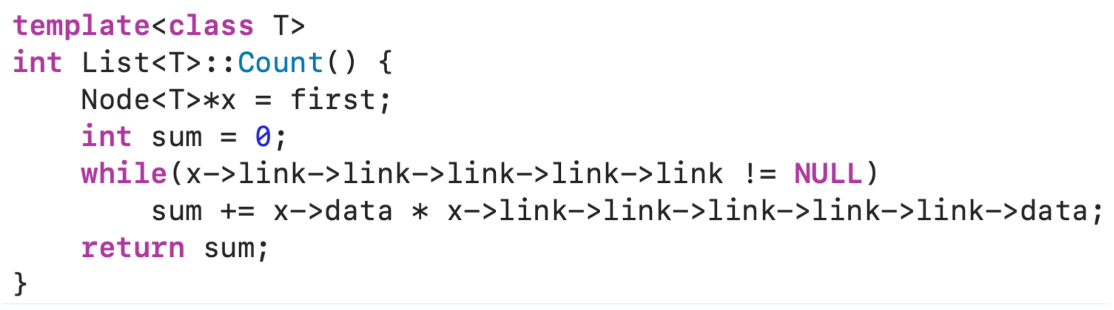
學號：110060007
姓名：黃俊穎

(b)

```
class List<T> { // add friend class for privite members if List
    ......
    friend class Stack<T>;
    ......
}

template<class T>
Class Queue:public List<T>{
    public:
        Queue();
        void Pop();
        T& Rear();
        T& Front();
        void Push(Const T& element);
        bool IsEmpty();
    private:
    Node<T> *front;
    Node<T> *rear;
};

template<class T>
void Queue<T>::Pop(){
    if(IsEmpty())
        throw "Queue is empty";
    else
        Node *x = front;
    front = front->link;
    delete x;
}

template<class T>
void Queue<T>::Push(const T element){
    Node *tmp = new Node;
    tmp->data = element;
    tmp->link = NULL;
    top = tmp;
    if(IsEmpty())
        rear = front = tmp;
    else {
        rear->link = tmp;
        rear = tmp;
    }
}

template<class T>
T Queue<T>::Rear(){
    return rear->data;
}

template<class T>
bool Stack<T>::Front(){
    return Front->data;
}

template<class T>
bool Queue<T>::IsEmpty() {
    return (rear == NULL && front == NULL);
}
```

學號：110060007
姓名：黃俊穎

(c)



```
template<class T>
int List<T>::Count() {
    Node<T>*x = first;
    int sum = 0;
    while(x->link->link->link->link->link != NULL)
        sum += x->data * x->link->link->link->link->link->data;
    return sum;
}
```