Question 1 (2.5 points):

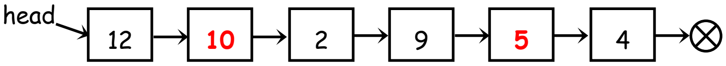
Given the definition of a singly linked list containing integers as follow:

```
struct Node{
     int data;
     Node* next;
};
```

```
struct List{
     Node* head;
};
```

Write function `int Count(List ls)` to count the number of node (data) satisfying that it is **equal to** |the next - previous node| in a linked list `ls` – **ignore** the first & the last node

For example: `ls is`

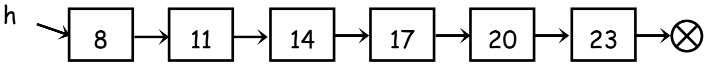


```
Count(ls) will return 2
```

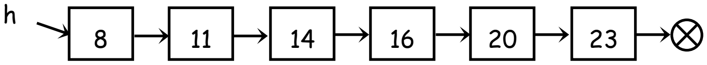
Question 2 (2.5 points):

Write a **recursion** function `bool checkIncrease(Node* h)` to check if a linked list **increases steadily**

For example: `checkIncrease(h) will return true`



`checkIncrease(h) will return false`



Question 3 (2.5 points):

A binary file named POLY.BIN stores the polynomials of a single indeterminate x as follow:

	Byte Order **(Thứ tự byte)**	**Size** **(Kích thước)**	**Description** **(Mô tả)**
Line #0	00:	**2** bytes **8** bytes **8** bytes	- Quantity of different terms of a polynomial (**e.g., n = 2**) - <deg$_1$><coef$_1$>: degree & coefficient of the 1st monomial - <deg$_2$><coef$_2$>: degree & coefficient of the 2nd monomial
Line #1	18:	**2** bytes **8** bytes **8** bytes **8** bytes	- Quantity of different terms of a polynomial (**e.g., n = 3**) - <deg$_1$><coef$_1$>: degree & coefficient of the 1st monomial - <deg$_2$><coef$_2$>: degree & coefficient of the 2nd monomial - <deg$_3$><coef$_3$>: degree & coefficient of the 3rd monomial
Line #3	43:	**2** bytes **8** bytes **8** bytes **8** bytes **8** bytes	- Quantity of different terms of a polynomial (**e.g., n = 4**) - <deg$_1$><coef$_1$>: degree & coefficient of the 1st monomial - <deg$_2$><coef$_2$>: degree & coefficient of the 2nd monomial - <deg$_3$><coef$_3$>: degree & coefficient of the 3rd monomial - <deg$_4$><coef$_4$>: degree & coefficient of the 4th monomial
…	…	…	…

Write function `readPolynomials()` to **read the polynomials of all lines** in this binary file, and function `writeGreatesDegreePoly()` to **write** to another binary file containing the polynomials with greatest degree (This file is named POLY_GREATES_DEG.BIN with the same format with POLY.BIN).

For example:

- The 1st polynomial $7x^3 + 7$ has two terms $7x^3$ and $7x^0$.

- The 2st polynomial $x^2 + x + 8$ has three terms x^2, x^1 and $8x^0$.

- The 3rd polynomial $2x^3 + x^2$ has two terms $2x^3$ and x^2.

We conclude the 1st and 3rd polynomials are the ones with greatest degree (3 in this case)

Question 4 (2.5 points):

Write function `void extractFactors(unsigned int A[], unsigned int N, unsigned int X, unsigned int*& B, unsigned int& M)` to extract the elements being the factors of X in array A with N integers into array B (dynamically allocated) with M elements just extracted from array A.

For example: `A = { 14, 2, 1, 13, 4 }  N = 5     X = 52`

`=> Result: B = {2, 13, 4}, M = 3 (Because 52 mod 2 = 52 mod 4 = 52 mod 13 = 0)`