Bai 1:

ho chương trình khởi tạo, trong đó:  
- struct node: đại diện cho một node của linked list  
- Hàm print: nhận vào con trỏ head của linked list và in ra từng phần tử của linked list  
- Hàm createLinkedList:  
    + Nhận vào số phần tử (>0) của link list  
    + Xây dựng một linked list với dữ liệu của các node được nhập từ đầu vào chuẩn (stdin), node mới được thêm vào **CUỐI**linked list  
    + Trả về con trỏ đến node đầu tiên của linked list.  
- Hàm evenThenOddLinkedList:  
    + Nhận vào con trỏ head của linked list  
    + Hám có tác vụ thay đổi Linked List đầu vào, sao cho các phần tử có giá trị chẵn nằm ở đầu linked list và các phần tử có giá trị lẻ nằm  
ở phía sau. Hơn nữa, thứ tự của các giá trị chẵn và lẻ này không thay đổi so với Linked List đầu vào  
- Hàm main đọc vào số phần tử của linked list, gọi hàm createLinkedList để khởi tạo linked list, sau đó gọi hàm print để in ra các phần tử của linked list.  
Bạn hãy hoàn thành hàm **evenThenOddLinkedList**  
**Đầu vào:**Tất cả giá trị từ đầu vào chuẩn (stdin) đều có giá trị trong khoảng (0; 5000)  
**Đầu ra:**Thỏa yêu cầu bài toán  
**English version**Write a program where:  
- struct node: represents a node in linkedlists  
- functions createLinkedList:  
    + Receive the size of a linked list (>0)  
    + create a linked list with value from standard input (stdin) a new node is add to the  **END** of the linked list  
    + Return a pointer which points to the first node of the linked list.  
- Function evenThenOddLinkedList:  
    + Receive the linked list's head pointer  
    + The function changes the input Linked List so that the even value will be the first part of the linked list and the odd value will be the latter part. Furthermore, the order of each (even/odd) group is not changed.  
- Function main reads the size of the linked list, calls functions createLinkedList to initialize linked list, then calls function print to print the linked list.  
Complete functions **evenThenOddLinkedList**  
**Input:**All value from standard input (stdin) with value in (0; 5000)  
**Output:**satisfy the requirements

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 1 | 5  2 4 6 8 10 | 2 4 6 8 10  2 4 6 8 10 |

Answer:

#include <iostream>

using namespace std;

struct node

{

int data;

node \*next;

};

node \*createLinkedList(int n); // The implementation is provided implicitly

node\* evenThenOddLinkedList(node \*head)

{

if (head == nullptr || head->next == nullptr) {

return head;

}

node\* evenHead = nullptr;

node\* evenTail = nullptr;

node\* oddHead = nullptr;

node\* oddTail = nullptr;

// Phân tách các phần tử chẵn và lẻ vào danh sách tạm thời

while (head != nullptr) {

node\* nextNode = head->next;

if (head->data % 2 == 0) {

if (evenHead == nullptr) {

evenHead = head;

evenTail = head;

} else {

evenTail->next = head;

evenTail =head;

}

} else {

if (oddHead == nullptr) {

oddHead = head;

oddTail = head;

} else {

oddTail->next = head;

oddTail = head;

}

}

head = nextNode;

}

// Kết hợp danh sách chẵn và danh sách lẻ

if (evenHead != nullptr) {

evenTail->next = oddHead;

return evenHead;

} else {

return oddHead;

}

}

void print(node\* head)

{

while (head != nullptr)

{

cout << head->data << " ";

head = head->next;

}

cout << endl;

}

int main()

{

int n = 0;

cin >> n;

node \*head = createLinkedList(n);

print(head);

head = evenThenOddLinkedList(head);

print(head);

return 0;

}

Bai 2

Cho chương trình khởi tạo, trong đó:  
- struct node: đại diện cho một node của linked list  
- Hàm print: nhận vào con trỏ head của linked list và in ra từng phần tử của linked list  
- Hàm createLinkedList:  
    + Nhận vào số phần tử (>0) của link list  
    + Xây dựng một linked list với dữ liệu của các node được nhập từ đầu vào chuẩn (stdin), node mới được thêm vào **CUỐI**linked list  
    + Trả về con trỏ đến node đầu tiên của linked list.  
- Hàm mergeList:  
    + Nhận đầu vào là 2 con trỏ head1, head2 tới 2 linked list  
    + Hàm mergeList sẽ chỉnh sửa linked list thứ 1 sao cho linked list thứ 2 được thêm vào phía sau linked list thứ 1  
- Hàm main đọc vào số phần tử của linked list, gọi hàm createLinkedList để khởi tạo linked list, sau đó gọi hàm print để in ra các phần tử của linked list.  
Bạn hãy hoàn thành hàm **mergeList**  
**Đầu vào:**Tất cả giá trị từ đầu vào chuẩn (stdin) đều có giá trị trong khoảng (0; 5000)  
**Đầu ra:**Thỏa yêu cầu bài toán  
**English version**  
- struct node: represents a node in linkedlists  
- functions createLinkedList:  
    + Receive the size of a linked list (>0)  
    + create a linked list with value from standard input (stdin) a new node is add to the  **END** of the linked list  
    + Return a pointer which points to the first node of the linked list.  
- Function mergeList:  
    + receive 2 head1, head2 pointers of 2 linked lists  
    + Function mergeList will change the 1st linked list so that the 2nd linked list will be concatenated to the end of the 1st linked list  
- function main reads the linked list's size, calls function createLinkedList to initialize the linked list, then calls function print to print the linked list.  
Complete the functions **mergeList**  
**Input:**  
All the inputs from standard input (stdin) with value in (0; 5000)  
**Output:**  
Satisfy the requirements  
Write a program where:

**For example:**

| **Test** | **Input** | **Result** |
| --- | --- | --- |
| 1 | 5  1 3 5 7 9  2  1 2 | 1  3  5  7  9  1  2 |