

Data Structure

Mid Term Exam

Contest Hints

Remove Duplicate Again

Explanation:

আপনাকে একটি Linked List দেয়া হবে। ঐ Singly Linked List এ যেসব নান্দ্রার গুলা Duplicate অর্থাৎ একের অধিক আছে তা Delete করে দিতে হবে। এবং সবশেষে ঐ Linked List টি Ascending order এ প্রিন্ট করে দেখাতে হবে।

Note : এই ক্ষেত্রে অবশ্যই **STL List** ব্যবহার করতে হবে , নাহলে মার্কস পাবেন না।

Hints

আমরা আগের এক্সামে দেখেছি , কীভাবে একটি Singly Linked List এর মধ্যে খুব সহজে ডুপ্লিকেট নান্দ্রার Delete এর বিষয়টি বের করেছি $O(N^2)$ কমপ্লেক্সিটি এর সাহায্যে। এই প্রবলেমের ক্ষেত্রেও আমরা লিস্ট থেকে ডুপ্লিকেট ভ্যালু গুলা খুজে বের করে ডিলিট করে দিতে হবে। এই ক্ষেত্রে STL List এর মেথড গুলা বিবেচনা করে দেখতে পারি কোন মেথড টি ব্যবহার করলে কাজ গুলা করা সহজ হবে । অথবা Ascending অর্থাৎ ছোট থেকে বড় আকারে সাজাতে হবে , তাই আমরা অপারেশন গুলা করার আগে বা পরে List টিকে সাজিয়ে নিতে পারি।

Palindrome

Explanation:

আপনাকে একটি ইন্টিজার ভ্যালুর Doubly Linked List দেয়া হবে। আপনাকে বলতে হবে এই Doubly Linked List এর ভ্যালু গুলো Palindrome আকারে আছে কিনা। অর্থাৎ বাম থেকে ডানে এবং ডান থেকে বামে একই ভাবে সাজানো আছে কিনা। যদি হ্যাঁ হয় তবে “YES” প্রিন্ট করতে হবে, না হলে “NO” প্রিন্ট করতে হবে।

Note : এই ক্ষেত্রে অবশ্যই **Doubly Linked List** ব্যবহার করতে হবে, নাহলে মার্কস পাবেন না।

Hints

আমরা সি প্রোগ্রামিং কোর্সে শিখে এসেছি কীভাবে খুব সহজে একটি স্পেশাল টেকনিকের সাহায্যে একটি array, Palindrome কিনা তা বের করা যায়। Doubly Linked List এ যেহেতু বাম থেকে ডানে, এবং একই সাথে ডান থেকে বামে traverse করা যায়, তাই এই প্রবলেমটি solve ক্ষেত্রে উক্ত টেকনিক টি ব্যবহার করা যাবে। এই ক্ষেত্রে আমরা Doubly Linked List টিতে একটি নির্দিষ্ট কন্ডিশন পর্যন্ত traverse করে খুব সহজে বলে দিতে পারবো লিস্ট টি Palindrome কিনা।

Browser History

Explanation:

আপনাকে একটি string এর Doubly Linked List দেয়া হবে যা হচ্ছে মূলত কিছু ওয়েবসাইট। মনে করুন আপনি একটি ব্রাউজার ব্যবহার করছেন এবং সেখানে কিছু History দেয়া আছে। এখন শুরুতে আপনি থাকবেন List এর প্রথম ওয়েবসাইটে।

এরপর আপনাকে ঐ লিস্টের উপর কাজ করার জন্য কিছু কুয়েরি দেয়া হবে। কুয়েরি গুলো নিচে উল্লেখ করা হলো।

১. **visit address** : এই অপারেশনের জন্য আপনাকে visit কমান্ড এর পাশাপাশি একটি website এর নাম দেয়া হবে। যেমনঃ “visit phitron”। তখন আপনাকে list এর মধ্যে যদি phitron থাকে তবে সেই পেজে যেয়ে phitron প্রিন্ট করতে হবে। যদি list এ phitron না পেয়ে থাকেন তবে প্রিন্ট করতে হবে “Not available”

২. **next** : next অপারেশন করতে বলা হলে আপনি এখন লিস্ট এর যে website এ আছেন তার পরবর্তী website এ যাবেন এবং সেই website টি প্রিন্ট করবেন। যদি সেই পজিশনের পরের পজিশনে ওয়েবসাইট না থাকে তবে সেই ক্ষেত্রে প্রিন্ট করতে হবে “Not available”

৩. **prev** : next অপারেশন করতে বলা হলে আপনি এখন লিস্ট এর যে website এ আছেন তার পূর্ববর্তী website এ যাবেন এবং সেই website টি প্রিন্ট করবেন। যদি সেই পজিশনের আগের পজিশনে ওয়েবসাইট না থাকে তবে সেই ক্ষেত্রে প্রিন্ট করতে হবে “Not available”

Hints

এই প্রবলেমটি মূলত **Linked List** এ **traversal** করার অর্থাৎ প্রত্যেকটি **element** কে এক্সেস করতে পারার একটি **simple** প্রবলেম। এই প্রবলেমটি সলভ করার ক্ষেত্রে সাজেশন থাকবে প্রথমে **string** এর **Doubly Linked** ব্যবহার করে সহজে প্রবলেম টি সলভ করার। এই ক্ষেত্রে আমরা **head** কে পয়েন্ট করা একটি পয়েন্টার রেখে খুব সহজে বামে ডানে মুভ করে এই প্রবলেমটি সলভ করে নিতে পারি।।

কি হবে যদি ভিজিট এড্রেস এর আগে পিছে নাল থাকে বা যেই নোড কে সার্চ করছি সেটা পুরো লিস্ট এই না পাওয়া যায়। এই কর্নার কেস গুলো চিন্তা করে ব্যাপারটা হ্যান্ডেল করার চেষ্টা করো

এবং পরবর্তীতে মার্কস পেয়ে যাওয়ার পর পরবর্তীতে **STL List** এর সাহায্যে প্রবলেম টি আবার সলভ করার চেষ্টা করতে পারি। এতে করে আমাদের **Raw** মেথড এর পাশাপাশি **built-in** জিনিস গুলো সহজে ব্যবহার করতে পারার স্কিল বাড়বে।

Queries Again

Explanation:

মনে করুন, আপনার কাছে একটি Doubly Linked List এ আছে যা শুরুতে খালি অবস্থায় আছে। এরপর আপনাকে কিছু কুয়েরি দেয়া হলো।

কুয়েরি:

একটি Index ও একটি Value দেয়া থাকবে। আপনাকে Doubly Linked List টির ঐ Index এ ভ্যালুটি Insert করতে হবে।

Index টি যদি লিস্টে না থাকে তবে সেই ক্ষেত্রে “Invalid” প্রিন্ট করতে হবে

প্রতিটি successful অপারেশন শেষে ঐ Updated Linked List টির ইলিমেন্ট গুলো প্রথমে বাম থেকে ডানে এবং পরবর্তীতে ইলিমেন্ট গুলো ডান থেকে বামে এই আকারে প্রিন্ট করতে হবে।

Note : এই ক্ষেত্রে অবশ্যই **Doubly Linked List** ব্যবহার করতে হবে , নাহলে মার্কস পাবেন না।

Hints

Doubly Linked List এর মডিউলে দেখানো হয়েছে কীভাবে **Doubly Linked List** এর যেকোন পজিশনে ভ্যালু Insert করা যায়। এই প্রবলেম এ ঠিক একই অপারেশন করতে বলা হয়েছে। **Doubly Linked List** এর **Module** এবং **Conceptual session** গুলো ঠিক মতো রিভিশন করলে **Doubly Linked List** এ ভ্যালু Insert করার পাশাপাশি **Doubly Linked List** এর ভ্যালু গুলোকে বাম থেকে ডানে এবং ডান থেকে বামে প্রিন্ট করা খুব সহজে করে ফেলা যাবে। প্রিন্ট করার ক্ষেত্রে আমরা প্রতিটি কুয়েরি তে **List** টি উল্টো প্রিন্ট করার পরিবর্তে যদি প্রতিক্ষেত্রে **Reverse** করে তারপর প্রিন্ট করি, সেই ক্ষেত্রে প্রবলেমটি **Accept** হবে কিনা চিন্তা করেন।

Remember Previous Queries?

Explanation:

মনে করুন, আপনার কাছে একটি Doubly Linked List দেয়া আছে যা শুরুতে খালি অবস্থায় আছে। এরপর আপনাকে কিছু কুয়েরি দেয়া হলো।

কুয়েরি সমূহঃ

X=0 : এই ক্ষেত্রে একটি ভ্যালু দেয়া হবে , ঐ ভ্যালুটি Linked List টির Head পজিশন Insert করতে হবে ।

X=1 : এই ক্ষেত্রে একটি ভ্যালু দেয়া হবে , ঐ ভ্যালুটি Linked List টির Tail পজিশনে Insert করতে হবে

X=2 : এই ক্ষেত্রে একটি Index দেয়া থাকবে , ঐ Index এর Node টি ডিলিট করে দিতে হবে। এইক্ষেত্রে Invalid Index দেয়া থাকলে কোনো অপারেশন করতে হবে না। শুধু মাত্র Linked List টি বাম থেকে ডানে এবং ডান থেকে বামে এই আকারে প্রিন্ট করে দিলেই হবে।

প্রতিটি successful অপারেশন শেষে ঐ Updated Linked List টির ইলিমেন্ট গুলো প্রথমে বাম থেকে ডানে এবং পরবর্তীতে ইলিমেন্ট গুলো ডান থেকে বামে এই আকারে প্রিন্ট করতে হবে ।

অপারেশন শেষে Linked List খালি থাকলে সেই ক্ষেত্রে L-> এবং R-> প্রিন্ট করে Newline প্রিন্ট করতে হবে। অর্থাৎ খালি প্রিন্ট হবে। (২য় টেস্টকেস)

Note : এই ক্ষেত্রে অবশ্যই **STL List** ব্যবহার করতে হবে , নাহলে মার্কস পাবেন না।

Hints

এই প্রবলেমটি মূলত **Doubly Linked** এর **STL** শেখার জন্য সেট করা হয়েছে। আমরা **STL List** এর মডিউলে দেখেছি কীভাবে একটি **List** এর **head** পজিশনে অর্থাৎ সবার শুরুতে , **tail** পজিশনে অর্থাৎ সবার শেষে খুব সহজে **List** এর **builtin** ফাংশন ব্যবহার করে ভ্যালু **Insert** করা যায়। এর পর দেখেছি **built-in** ফাংশন ব্যবহার করে কীভাবে খুব সহজে একটি **index** এর ভ্যালু **delete** করা যায়। এই ক্ষেত্রে সেই পজিশনের **iterator** পাওয়ার জন্য আমরা একটি বিশেষ ফাংশনের ব্যবহার করেছিলাম । উক্ত মডিউল গুলো রিভিশন করলে আশা করি খুব সহজে এই প্রবলেমটি সল্ভ করা যাবে

