#### **Algorithm**

**Final Exam**

**Contest Hints**

| **Tetranacci Number I** |
| --- |
| **Explanation:**  এই সমস্যাটিতে আপনাকে Tetranacci সিকোয়েন্সের n-তম পদটি বের করতে হবে। Tetranacci সিকোয়েন্স হলো Fibonacci সিকোয়েন্সের একটি এক্সটেনশন, যেখানে প্রতিটি পদ আগের চারটি পদের যোগফল দ্বারা নির্ধারিত হয়। Tetranacci সিকোয়েন্সটি নিম্নরূপে সংজ্ঞায়িত করা হয়েছে:  **T₀ = 0, T₁ = 1, T₂ = 1, T₃ = 2**  **n ≥ 4 এর জন্য, Tₙ = Tₙ₋₁ + Tₙ₋₂ + Tₙ₋₃ + Tₙ₋₄**  আপনাকে একটি ইন্টিজার n দেওয়া হবে, এবং আপনাকে Tetranacci সিকোয়েন্সের n-তম পদটি রিটার্ন করতে হবে।  **Note:** এই সমস্যাটি **Recursion (Top Down)** পদ্ধতি ব্যবহার করে সমাধান করতে হবে।  **Hints**   | আপনি প্রথমে N ইনপুট নিবেন। তারপর একটা ফাংশন লিখবেন যা int রিটার্ন করবে। সেই ফাংশন এর প্যারামিটার হিসেবে N কে পাঠাবেন। তারপর ফাংশনে নিচের ধাপগুলি অনুসরণ করবেন -  **১. বেস কেস নির্ধারণ করুন:**  \* যদি n = 0 হয়, তাহলে T₀ = 0 রিটার্ন করবেন  \* যদি n = 1 হয়, তাহলে T₁ = 1 রিটার্ন করবেন  \* যদি n = 2 হয়, তাহলে T₂ = 1 রিটার্ন করবেন  \* যদি n = 3 হয়, তাহলে T₃ = 2 রিটার্ন করবেন  **২. রিকার্শন স্টেপ:**  যদি n >= 4 হয়, তাহলে **Tₙ = Tₙ₋₁ + Tₙ₋₂ + Tₙ₋₃ + Tₙ₋₄** রিকার্শন ব্যবহার করে গণনা করবেন।  **৩. ফাইনাল মান রিটার্ন করুন:**  রিকার্শন সম্পূর্ণ হলে, **Tₙ** এর মান রিটার্ন করুন। | | --- | |

| **Tetranacci Number II** |
| --- |
| **Explanation:**  এই সমস্যাটিতে আপনাকে Tetranacci সিকোয়েন্সের n-তম পদটি বের করতে হবে। Tetranacci সিকোয়েন্স হলো Fibonacci সিকোয়েন্সের একটি এক্সটেনশন, যেখানে প্রতিটি পদ আগের চারটি পদের যোগফল দ্বারা নির্ধারিত হয়। Tetranacci সিকোয়েন্সটি নিম্নরূপে সংজ্ঞায়িত করা হয়েছে:  **T₀ = 0, T₁ = 1, T₂ = 1, T₃ = 2**  **n ≥ 4 এর জন্য, Tₙ = Tₙ₋₁ + Tₙ₋₂ + Tₙ₋₃ + Tₙ₋₄**  আপনাকে একটি ইন্টিজার n দেওয়া হবে, এবং আপনাকে Tetranacci সিকোয়েন্সের n-তম পদটি রিটার্ন করতে হবে।  **Note:** এই সমস্যাটি **Loop (Bottom up or Tabulation)** পদ্ধতি ব্যবহার করে সমাধান করতে হবে।  **Hints**   | এই সমস্যা সমাধানে আমরা **Loop বা Tabulation বা Bottom-Up** ডাইনামিক প্রোগ্রামিং পদ্ধতি ব্যবহার করবেন। নিচে ধাপে ধাপে ব্যাখ্যা করা হলো:  ১. প্রথমে n ইনপুট নিবেন, যা Tetranacci সিকোয়েন্সের n-তম পদ বের করতে ব্যবহৃত হবে।  ২. তারপর, আমরা একটি অ্যারে ডিক্লেয়ার করবো, যার সাইজ n + 1 । এই অ্যারেটি Tetranacci সিকোয়েন্সের মানগুলি স্টোর করবে।  ৩. এরে এর ০ থেকে ৩ নং ইনডেক্স পর্যন্ত এর মানগুলি বেস কেস হিসেবে ইনিশিয়ালাইজ করবো:  arr[0] = 0  arr[1] = 1  arr[2] = 1  arr[3] = 2  ৪. লুপ ব্যবহার করে মানগুলি ক্যালকুলেট করবো। লুপ i, 4 থেকে n পর্যন্ত চালাতে হবে। যেখানে প্রতিটি i এর জন্য, নিম্নলিখিত সূত্র ব্যবহার করে arr[i] এর মান ক্যালকুলেট করা:  arr[i]=arr[i−1]+arr[i−2]+arr[i−3]+arr[i−4]  (এই সূত্রটি Tetranacci সিকোয়েন্সের সংজ্ঞা অনুযায়ী আগের চারটি পদের যোগফল দ্বারা বর্তমান পদটি ক্যালকুলেট করে।)  ৫. লুপ শেষ হলে, arr[n] এর মান প্রিন্ট করা, যা Tetranacci সিকোয়েন্সের n-তম পদ আউটপুট দেখাবে।  এক্ষেত্রে N তম Tertranacci সংখ্যার মান কত বড় হতে পারে তার উপর বেইস করে ডাটা টাইপ সিলেকশনে সতর্ক থাকতে হবে | | --- | |

| **Make It** |
| --- |
| **Explanation:**  আপনাকে একটা ইন্টিজার N দেওয়া হবে। আপনি ১ থেকে শুরু করে নিচের দুইটা স্টেপ যতবার ইচ্ছা ততবার ফলো করে N বানানোর চেষ্টা করবেন।  ১। কারেন্ট ভ্যালু এর সাথে ৩ যোগ করে  ২। কারেন্ট ভ্যালু এর সাথে ২ গুণ করে |
| আপনাকে বলতে হবে আপনি N বানাতে পারবেন কিনা?  **Hints**   | প্রথমে Test case সংখ্যা ইনপুট নিব, তারপর প্রতিটি Test case এর জন্য, আপনি প্রথমে N ইনপুট নিবেন। তারপর একটা ফাংশন লিখবেন যা boolean রিটার্ন করবে। সেখানে ১ কে পাঠাবেন। যদি কারেন্ট ভ্যালু N এর সমান হয় তাহলে true রিটার্ন করবে। যদি N এর থেকে বড় হয় তাহলে false রিটার্ন করবে। আর নাহলে চেক করবে dp তে কারেন্ট ভ্যালু ক্যাল্কুলেটেড আছে কিনা। যদি থাকে তাহলে সেটা রিটার্ন করবে। আর নাহলে সেই ফাংশনকে একবার কারেন্ট ভ্যালু এর সাথে ৩ যোগ করে কল কররে দেখবে আর একবার কারেন্ট ভ্যালু এর সাথে ২ গুন করে দেখবে। তারপর তাদের রিটার্ন করা ভ্যালু অথবা ডিপিতে সেইভ করে রিটার্ন করে দিবে। | | --- | |

| **Adventure** |
| --- |
| | **Explanation:** একবার, একটি গুপ্তধন শিকারী ছিল যে একটি মূল্যবান শিল্পকর্মের সন্ধানে একটি প্রাচীন মন্দিরে প্রবেশ করেছিল। মন্দিরটি ফাঁদ এবং বাধা দিয়ে পূর্ণ ছিল এবং গুপ্তধন শিকারীকে তার সমস্ত সরঞ্জাম তার সাথে নিয়ে যেতে হয়েছিল।  গুপ্তধন শিকারীর একটি সীমিত ওজন ক্ষমতা সহ একটি ব্যাকপ্যাক ছিল এবং তিনি তার সাথে শুধুমাত্র একটি নির্দিষ্ট পরিমাণ সরঞ্জাম বহন করতে পারতেন। প্রতিটি টুকরো সরঞ্জামের নিজস্ব ওজন এবং মান ছিল এবং মোট ওজনকে সীমার মধ্যে রেখে মোট মূল্য সর্বাধিক করার জন্য কোন আইটেমগুলি আনতে হবে তা বেছে নিতে ট্রেজার হান্টারকে প্রয়োজন ছিল।  ট্রেজার হান্টারকে তার ব্যাকপ্যাকে কোন আইটেম আনতে হবে তা বেছে নিতে হবে যাতে তার ব্যাকপ্যাকের মোট ওজন একটি নির্দিষ্ট সীমার মধ্যে রেখে তাদের মোট মূল্য সর্বাধিক করা যায়। প্রতিটি আইটেম শুধুমাত্র একবার অন্তর্ভুক্ত করা যাবে।.  **Hints** | | --- | | প্রথমে Test case সংখ্যা ইনপুট নিব, তারপর প্রতিটি Test case এর জন্য প্রথমে আইটেম সংখ্যা আর capacity ইনপুট নিব। তারপর প্রত্যেক আইটেম এর weight আর value ইনপুট নিবো। তারপর মডিউলে Dynamic Programming এর এরকম একটা টপিক দেখানো হয়েছে ঠিক সেইভাবে Bottom up কিংবা top down এপ্রোচ ফলো করবো। তাহলেই আমরা আন্সার পেয়ে যাবো। | |

| **Pile of Word**  **Explanation:** এই সমস্যাটিতে আপনাকে দুটি স্ট্রিং S1 এবং S2 দেওয়া হবে। আপনাকে চেক করতে হবে যে এই দুটি স্ট্রিং একে অপরের Pile of Word কিনা। যদি Pile of Word হয় তাহলে YES প্রিন্ট করতে হবে। অন্যথায়, NO প্রিন্ট করতে হবে।  Pile of Word হল এমন একটি শব্দ যা অন্য শব্দের অক্ষরগুলিকে পুনর্বিন্যাস করে সেই শব্দ তৈরী করা যায়। অন্য কথায়, এতে একটি নির্দিষ্ট স্ট্রিং S1 এর অক্ষরগুলিকে পুনর্বিন্যাস করে একটি নতুন স্ট্রিং S2 তৈরি করা।  অর্থাৎ, S1 এবং S2 এর অক্ষরগুলি একই কিনা এবং প্রতিটি অক্ষরের ফ্রিকোয়েন্সি (কতবার ব্যবহৃত হয়েছে) একই কিনা। |
| --- |
| **Hints**   | প্রথমে টেস্ট কেস সংখ্যা T ইনপুট নিন। তারপর, প্রতিটি টেস্ট কেসের জন্য S1 এবং S2 ইনপুট নিন।  **১. দুইটি স্ট্রিং এর দৈর্ঘ্য চেক করা:** যদি S1 এবং S2 এর দৈর্ঘ্য একই না হয়, তাহলে অবশ্যই তারা Pile of Word নয়। সেক্ষেত্রে NO প্রিন্ট করুন এবং পরবর্তী টেস্ট কেসে চলে যান।  **২.** দুইটি স্ট্রিং এর জন্য দুটি ফ্রিকোয়েন্সি অ্যারে frq1 এবং frq2 তৈরি করুন, যেখানে প্রতিটি অ্যারে 26টি এলিমেন্ট থাকবে (ইংরেজি ছোট হাতের অক্ষরের জন্য)। S1 এবং S2 এর প্রতিটি অক্ষরের ফ্রিকোয়েন্সি গণনা করুন এবং সংশ্লিষ্ট ফ্রিকোয়েন্সি অ্যারেতে স্টোর করুন।  **৩.** দুটি ফ্রিকোয়েন্সি অ্যারে তুলনা করুন। যদি সব অক্ষরের ফ্রিকোয়েন্সি একই হয়, তাহলে YES প্রিন্ট করুন। অন্যথায়, NO প্রিন্ট করুন। | | --- | |