y in ○ শাফায়েতের ব্লগ প্রোগ্রামিং, অ্যালগরিদম, ব্যাকএন্ড ইঞ্জিনিয়ারিং আমার সম্পর্কে মডুলার অ্যারিথমেটিক শাফায়েত, সফটওয়্যার ইঞ্জিনিয়ার @ Meta (বিস্তারিত...) 🛗 जानूग्राति २०, २०১२ by भाषाराज কম্পিউটার বিজ্ঞান সিরিক ⁹৯০ গ্রাফ অ্যালগরিদম শাফায়েত আশরাফ -১৭ কে ৫ দিয়ে ভাগ করলে ভাগশেষ কত হয়? ২ 5000 কে ১৭ দিয়ে ভাগ করলে ভাগশেষ কত হয় সেটা কি তুমি ওভারফ্লো এড়িয়ে নির্ণয় করতে পারবে? O(n) এ পারলে $O(\log_2 n)$ কমপ্লেক্সিটিতে পারবে? যদি কোনো একটি উত্তর "না" হয় তাহলে এই পোস্ট তোমার জন্য। তবে তুমি যদি মডুলার ইনভার্স বা এডভামড কিছু শিখতে পোস্টটি খুলো তাহলে তোমাকে আপাতত হতাশ করতে হচ্ছে। সি/জাভা সহ বেশিভাগ প্রোগ্রামিং ল্যাংগুয়েজে এ % কে ভাগশেষ অপারেটর ধরা হয়। x কে m দিয়ে ভাগকরে ভাগশেষ বের করার অর্থ x%m এর মান বের করা অথবা আমরা বলতে পারি x কে m দিয়ে mod করা। "determine answer modulo 1000" এ কথাটির অর্থ হলো উত্তরকে ১০০০ দিয়ে mod করে তারপর আউটপুট দিতে হবে। একটি সমস্যা দিয়ে শুরু করি। তোমার ১০০টি বই আছে,তুমি কয়ভাবে বইগুলো সাজাতে পারবে? খুব সহজ,১০০! (১০০ ফ্যাক্টরিয়াল) ভাবে সাজাতে পারবে। ১০০! ১৫৮ ডিজিটের বিশাল একটি সংখ্যা। তাই আমি তোমাকে প্রবলেমটা সহজ করে দিলাম,ধরো তুমি x উপায়ে বইগুলো সাজাতে পারবে,তাহলে তোমাকে x কত সেটা বলতে হবে। অর্থাত ১০০! বের করে ৯৭ দিয়ে ভাগ করে ভাগশেষটা বের করাই তোমার সমস্যা। (Determine 100 factorial modulo 97) প্রোগ্রামিং কনটেস্ট এবং অ্যালগোরিদম এটা কিভাবে করবে? ১০০! এর মান তুমি বের করতে পারবেনা ৬৪বিট আনসাইনড ইণ্টিজার দিয়েও, এরা ২^{৬৪} — ১ পর্যন্ত সংখ্যা নিয়ে কাজ করতে পারে, তাই ওভারফ্লো হবে। কিন্তু আমরা জানি আমাদের উত্তর অনুপ্রেরণা(৪): কখনোই 97 এর বড় হরেনা কারণ **কোনো সংখ্যাকে m দিয়ে mod করা হলে সংখ্যাটি m এর থেকে বড় হতে পারবেনা**। কেন আমি প্রোগ্রামিং শিখবো? কম্পিউটার বিজ্ঞান কেন পডবো? আমরা এ ধরণের সমস্যা সমাধান করতে সাহায্য নিবো দুটি সুত্রের: প্রোগ্রামিং কনটেস্ট এবং অনলাইন জাজে হাতেখড়ি কনফিউজড প্রোগ্রামার (a+b)%m = ((a%m) + (b%m))%m(a*b)%m = ((a%m)*(b%m))%mব্যাকএন্ড ইঞ্জিনিয়ারিং: ডিস্ট্রিবিউটেড ফাইল সিস্টেম n সংখ্যক নম্বর $a_1, a_2 \dots a_n$ এর জন্য সুত্র দুটি ব্যবহার করতে পারবে। ৰুম ফিল্টার লগ-স্ট্রাকচার্ড-ট্রি উপরের সমস্যাটিতে ২য় সুত্রটি লাগবে। তোমার বের করা দরকার ১০০! %৯৭ অর্থাত: কনসিস্টেন্ট হ্যাশিং(নতুন) (>00*৯৯*৯৮*....*১)%৯৭ ক্যাশিং অ্যালগরিদম: এল-আর-ইউ ক্যাশ তুমি যেটা করবে সেটা হলো গুণ করার সময় ২য় সুত্রের মত করে mod করতে থাকবে,তাহলে কোনো সময়ই overflow ঘটবেনা কারণ mod করলে প্রতি স্টেপে সংখ্যাটি ছোটো হয়ে যাচ্ছে। এটার কোড হতে পারে এরকম: অ্যালগরিদম বেসিক(৬): বিগ "O" নোটেশন 1 int fact=1; for(int i=1;i<=100;i++) কমপ্লেক্সিটি ক্লাস(P-NP, টুরিং মেশিন ইত্যাদি) fact=((fact%97)*(i%97))%97; হান্টিং প্রবলেম वारेताति जार्চ - ১ printf("%d\n",fact); वारेताति जार्চ - २(वारेटजकभत) এটার আউটপুট আসবে ০। অর্থাত ১০০! % ৯৭ =০। একটু খেয়াল করলেই বুঝবে এখানে আমরা ২য় সুত্রটি প্রয়োগ করেছি ২টি করে সংখ্যা নিয়ে। ফুয়েড সাইকেল ফাইন্ডিং অ্যালগোরিদম সূত্র দুটি কেনো কাজ করে সেটা জানা দরকার। আমি ১ম সুত্রটির প্রমাণ দেখাচ্ছি,২্যাটিও একইভাবে করা যায়। প্রমানটি আমার নিজের মত করে করা। ডাটা স্ট্রাকচার(১৩): লিংকড লিস্ট ধরি (x+y)%৫ এর মান আমাদের বের করতে হবে। এখন যদি $x\%5=c_1$ আর $y\%5=c_2$ হয়,তাহলে x কে আমরা লিখতে পারি $5n_1+c_1$ এবং y কে লিখতে পারি $5n_2+c_2$ যেখানে n_1 আর n_2 দুটি স্ট্যাক ইন্টিজার। এটা একদম বেসিক রুল,আশা করে বুঝতে সমস্যা হচ্ছেনা। এখন: কিউ+সার্কুলার কিউ (x+y)%5শ্লাইডিং রেঞ্জ মিনিমাম কুয়েরি (ডিকিউ) ডিসজয়েন্ট সেট(ইউনিয়ন ফাইন্ড) $=(5n_1+c_1+5n_2+c_2)\%5$ ট্রাই(প্রিফিক্স ট্রি/রেডিক্স ট্রি) $= (5n_1 + 5n_2 + c_1 + c_2)\%5$ ——(5) সেগমেন্ট ট্রি-১ সেগমেন্ট ট্রি-২(লেজি প্রপাগেশন) এখানে $5n_1+5n_2$ অবশ্যই 5 এর মাল্টিপল,তাই আমরা লিখতে পারি অ্যারে কমপ্রেশন/ম্যাপিং $5n_1+5n_2=5N$ যেখানে $N=n_1+n_2$ লোয়েস্ট কমন অ্যানসেস্টর এবং $c_1+c_2=C$ বাইনারি ইনডেক্সড টি তাহলে (১) থেকে পাচ্ছি: স্কয়ার-ক্রট ডিকম্পোজিশন (5N+C)%5গ্রাফ থিওরি(২০): গ্রাফ থিওরিতে হাতেখড়ি এখন পরিস্কার বোঝা যাচ্ছে যে উত্তর হলো C%5। C কে আবার mod করতে হলো কারণ c_1+c_2 এর মান 5 এর থেকে বড় হতেই পারে। এখন অ্যাডজেসেন্সি ম্যাট্রিক্স অ্যাডজেসেন্সি লিস্ট **44** ((x%5) + (y%5))%5——(२) ব্রেথড ফার্স্ট সার্চ (বিএফএস) =((5n1+c1)%5)+((5n2+c2)%5))%5মিনিমাম স্প্যানিং ট্রি ১ (প্রিমস অ্যালগোরিদম) $(5n_1+c_1)\%5=c_1$ মিনিমাম স্প্যানিং ট্রি ২ (ক্রুসকাল অ্যালগোরিদম) টপোলজিকাল সর্ট $(5n_2+c_2)\%5=c_2$ ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলোজিকাল সর্ট ডায়াক্সট্টা তাহলে ২ কে লিখতে পারি: ফুয়েড ওয়ার্শল $(c_1+c_2)\%5=C\%5$ বেলম্যান ফোর্ড তাহলে ১ম সুত্রটি প্রমাণিত হলো। তারমান যোগ করে mod করা আর আগে mod করে তারপর যোগ করে আবার mod করা একই কথা। সুবিধা হলে সংখ্যাটি কোনো স্টেপেই বেশি বড় হতে পারেনা। গুণের ক্ষেত্রেই আর্টিকুলেশন পয়েন্ট এবং ব্রিজ একই সুত্র প্রযোজ্য। স্ট্রংলি কানেক্টেড কম্পোনেন্ট ম্যাক্সিমাম ফ্রো-১ নেগেটিভ সংখ্যার \mod নিয়ে একটু আলাদা ভাবে কাজ করতে হয়। সি তে -17%5 এর মান দেখায় -২। কিন্তু সচরাচর আমরা ভাগশেষের যে সংজ্ঞা ব্যবহার করি তাতে x%m=p হলে গাণিতিকভাবে ম্যাক্সিমাম ফ্রো-২ স্টেবল ম্যারেজ প্রবলেম $m{u}$ m এর সবথেকে বড় থেকে বড় মাল্টিপল যেটা x এর থেকে ছোট সেই সংখ্যাটিকে x থেকে বিয়োগ করলে যে সংখ্যাটি পাওয়া যায় সেটাই p pঅয়লার ট্যুর ট্রি এর ডায়ামিটার নির্ণয় লংগেস্ট পাথ প্রবলেম যেমন 23%5 এর ক্ষেত্রে ৫imes ৪= ২০ হলো ৫ এর সবথেকে বড় মাল্টিপল যেটা ২৩ এর থেকে ছোট,তাই 23%5=23-(5 imes4)=3। -17%5 এর ক্ষেত্র খেয়াল করো -20 হলো ৫ এর সবথেকে বড় মাল্টিপল যেটে —১৭ থেকে ছোট,তাই উত্তর হবে ৩। অ্যালগরিদম গেম থিওরি(৩): এই কেসটা handle করা একটি উপায় হলো নেগেটিভ সংখ্যাটিকে একটি 5 এর মান্টিপল এর সাথে যোগ করা যেন সংখ্যাটি ০ থেকে বড় হয়ে যায়,তারপরে mod করা। যেমন: গেম থিওরি-১ গেম থিওরি-২ **66** -17%5 গেম থিওরি-৩ =(-17+100)%5 ডাইনামিক প্রোগ্রামিং: =83%5 ডাইনামিক প্রোগ্রামিং ১ (ফিবোনাচ্চি) =3 ডাইনামিক প্রোগ্রামিং ২ (শর্টেস্ট পাথ) ডাইনামিক প্রোগ্রামিং ৩ (LIS এবং পাথ প্রিন্টিং) এটা উপরের সুত্রের প্রমাণের মত করেই কাজ করে,একটু গুতালেই প্রমাণ করতে পারবে। **নেগেটিভ সংখ্যার mod নিয়ে কনটেস্টে সবসময় সতর্ক থাকবে,এটা wrong answer খাওয়ার একটা বড় কারণ হতে পারে।** ডাইনামিক প্রোগ্রামিং ৪ (লংগেস্ট কমন সাবসিকোয়েন্স) এবার আসি সুপরিচিত big mod সমস্যায়। সমস্যাটি হলো তোমাকে $(a^b)\%m$ এর মান বের করতে হবে, a,b,m তোমাকে বলে দেয়া হবে,সবগুলোর range 2^{31} পর্যন্ত হতে পারে। ১০০! % ৯৭ বের করার মত ডাইনামিক প্রোগ্রামিং ৫ (কয়েন চেঞ্জ/০-১ ন্যাপস্যাক) করে সহজেই তুমি overflow না খেয়ে মানটি বের করতে পারবে,সমস্যা হলো তুমি লুপ চালিয়ে একটি একটি গুণ করে $2^{2000000000}$ বের করতে চাইলে উত্তর পেতে পেতে সম্ভবত নাস্তা শেষ করে আসতে পারবে। ডাইনামিক প্রোগ্রামিং ৬ (সাবসেট সাম, কদ্বিনেটরিক্স, ডিসিশন প্রবলেম) ডাইনামিক প্রোগ্রামিং ৭ (ম্যাট্রিক্স চেইন মাল্টিপ্লিকেশন) আমরা চাইলে $O(\log_2 n)$ এ এটা করতে পারি। ডাইনামিক প্রোগ্রামিং ৮ ((বিটমাস্ক, ট্রাভেলিং সেলসম্যান) লক্ষ করো ডাইনামিক প্রোগ্রামিং ৯ (ট্রি, মিনিমাম ভারটেক্স কভার) $11 2^{100}$ $=(2^{50})^2$ ব্যাকট্যাকিং: ব্যকট্ট্যাকিং বেসিক এবং পারমুটেশন জেনারেটর এবং নাম্বার থিওরি/গণিত: $=(2^{25})^2$ মডুলার অ্যারিথমেটিক প্রাইম জেনারেটর (Sieve of Eratosthenes) এখন বলো 2^{50} বের করতে কি 2^{26} , 2^{27} ইত্যদি বের করার দরকার আছে নাকি 2^{25} পর্যন্ত বের করে square করে দিলেই হচ্ছে? আবার 2^{25} পর্যন্ত আসতে $(2^{12})^2$ পর্যন্ত বের করে square করে সাথে ২ গুণ করে বিটওয়াইজ সিভ ডিরেঞ্জমেন্ট দিলেই যথেষ্ট,অতিরিক্ত ২ গুণ করছি সংখ্যাটি বিজোড় সে কারণে। প্রতি স্টেপে গুণ করার সময় mod করতে থাকবে যাতে overflow না হয়। recursion ব্যবহার করে কোডটি লেখা জলের মত সোজা: প্রোবাবিলিটি: এক্সপেক্টেড ভ্যালু 1 #define i64 long long 2 i64 M; 3 i64 F(i64 N,i64 P) শ্ট্রিং ম্যাচিং(২): if(P==0) return 1; রবিন-কার্প if(P%2==0)কেএমপি (KMP) i64 ret=F(N,P/2); return ((ret%M)*(ret%M))%M; অন্যান্য(৩) : else return ((N%M)*(F(N,P-1)%M))%M; 12 13 } ডিরেকশন অ্যারে মিট ইন দ্যা মিডল টেইল-কল রিকার্শন অপটিমাইজেশন<mark>(নতুন)</mark> মন্তব্য অংশে "হাসান" একটি বিগ মডের সুন্দর রিকার্শন-ট্রি এর ছবির লিংক দিয়েছে, ছবিটা এরকম: 2^100 কোয়ান্টাম কম্পিউটার কোয়ান্টাম কম্পিউটার কী? 2^50 * কোয়ান্টাম কম্পিউটারের শক্তি এবং সীমাবদ্ধতা 2^25 2^25 * 2 2^24 **English Blog** 2^12 2^12 * 2^6 2^6 * 2^3 * 2^3 2^2 2^1 2^1* 2 2^0 (Base case) মডুলার অ্যারিথমেটিক ব্যবহার করে বিশাল আকারের ফলাফল কে আমরা ছোট করে আনতে পারি ফলাফলে বিভিন্ন প্রোপার্টিকে নষ্ট না করে,তাই এটা গণিতে খুব গুরুত্বপূর্ণ। প্রোগ্রামিং কনটেন্টে প্রায়ই বিভিন্ন প্রবলেমে মডুলার অ্যারিথমেটিক প্রয়োজন পড়বে,বিশেষ করে counting আর combinatorics এ যেখানে ফলাফল অনেক বড় হতে পারে,ফ্যাক্টরিয়াল নিয়ে কাজ করতে হতে পারে। **ভাগ করার সময় গুণ,আর যোগের মত সূত্র দুটি কাজ করেনা**,এটার জন্য তোমাকে extended euclid আর modular inverse জানতে হবে। সিপিউর জন্য mod খুব costly একটা অপারেশন। **যোগ,গুণের থেকে mod করতে অনেক বেশি সময় লাগে।** অপ্রয়োজনে mod ব্যবহার করলে কোড time limit exceed করতে পারে,তাই overflow হবার আশংকা না থাকলে সব জায়গায় mod করা দরকার নেই। আমার একটি কোড ৩সেকেন্ডে time limit exceed হবার পর খালি কিছু mod সরিয়ে ১.৩ সেকেন্ড নামিয়ে এনেছি। এখন চিন্তা করার জন্য একটি প্রবলেম। ধরো তোমাকে একটি অনেক বড় সংখ্যা(bigint) দিয়ে সেটাকে ২ $^{\circ\circ}$ এর ছোট একটি সংখ্যা দিয়ে mod করতে বলা হলো। $O(length_of_bigint)$ কমপ্লেক্সিটিতে কিভাবে সাহায্য: \$\$\\\ \tag{0*\so+\tag{*\so}}\$\$ >२のか=(((o*>0+>)*>0+ ₹)*>0+の)*>0+か প্র্যাকটিসের জন্য প্রবলেম: http://uva.onlinejudge.org/external/3/374.html http://uva.onlinejudge.org/external/101/10127.html Print PDF ফেসবুকে মন্তব্য 0 comments **0 Comments** Sort by Oldest \$ Add a comment... Facebook Comments plugin Powered by Facebook Comments 🖿 Posted in অ্যালগরিদম/প্রবলেম সলভিং, গণিত, প্রোগ্রামিং 💮 ? Tagged গণিত, নাম্বার থিওরী, বিগ মড, মডুলার-অ্যারিথমেটিক 89,711 বার পড়া হয়েছে ব বিটওয়াইজ্ সিভ(Bitwise sieve) গ্রাফ থিওরিতে হাতেখড়ি ৮: ডেপথ ফার্স্ট সার্চ এবং আবারো টপোলোজিকাল সর্ট 🕨 34 thoughts on "सङ्जात व्यातिश्रसिक Raihan जूनe ১৫, २०১२ at ७:८२ pm i64 দিয়ে কি বুঝিয়েছেন? long long int ব্যবহার করবো? Reply শাফায়েত जुत्तе ५७, २०५२ at ५२:०७ pm i64 দিয়ে long long বুঝানো হয়েছে। Reply হাসান জুলाই २১, २०১२ at ৯:०५ pm ভাইয়া, ট্টি টা কি এই রকম হবে? https://picasaweb.google.com/271emtiaj/UntitledAlbum#5767638674915293362 Reply শাফায়েত जूलारे २२, २०১२ at ७:১৮ pm হ্যা এরকমই হবে,ভালো হয়েছে ছবিটা। Reply Sumit93 সেপ্টেম্বর ২৩, ২০১২ at 8:0৮ pm vai tree ta jodi apni ai page add kore den khub valo hobe.Bz oi link e giye tree ta dekha jacce na. Reply Sumit93 সেপ্টেম্বর ২৩, ২০১২ at 8:১০ pm tree ta page add korle vlo hoy Reply শাফায়েত तर्ভञ्चत ১२, २०১२ at ১०:১৯ pm 2^100 2^25 2^25 * 2 2^24 2^12 2^12 * 2^6 2^6 * 2^3 2^3 * 2 2^2 2^1 2^1* 2 2^0 (Base case) Reply Sumit জानूग्राति २२, २०১७ at ৫:8० pm শাফায়েত ভাই আপনি মডুলার ইনভার্স নিয়ে কিছু লিখলে ভালো হয়। [[কমেন্টটি বাংলায় লিখে দিলাম, ইংবেজি ফন্টে বাংলা কমেন্ট ভবিষ্যতে approve করা হবে না — শাফায়েত]] Reply শাফায়েত জानुग्राति २७, २०১७ at ७:२० pm চেষ্টা করবো তবে আমার লেখার অপেক্ষায় না থেকে একটু ঘাটাঘাটি করে শিখে ফেলো। Reply শিখর রায় रफक्रग्राति २১, २०১७ at ७:०৭ pm ভাইয়া ক্যালকুলেটর ইউচ করলে তো -১৭%৫= ২ আসে। মানে আমি করেছি এভাবে -১৭/৫ = -৩ সমস্ত ২ বাই ৫ । কিন্তু আপনি দ্যাখালেন -১৭%৫= ৩ । আমি আসলে এই ব্যাপারটা নিয়ে কনফিউসড । বিভাজ্যতার নিয়ম অনুযায়ী তো আপনি ঠিক আছেন, তাহলে ক্যালকুলেটর কি ভুল? নাকি দুটই ঠিক আছে, আমারি বুঝতে কোথাও প্রবেম হচ্ছে! Reply <u>্র</u> শাফায়েত रफ्ऋग्राति २১, २०১७ at ७:८९ pm আমিতো আমার উবুটুর ক্যালকুলেটরে "–17 mod 5" লিখে ৩ পাচ্ছি। Reply **Shikhor Kumer Roy** *रफ*ङ्मग्राति २১, २०১७ at २:১७ pm ভাইয়া এমনিতে ক্যালকুলেটর চেপে ভাগ করেন -১৭/৫ = -৩.৪ । আর -৩.৪ মানে তো "-৩ সমস্ত ২ বাই ৫" । অর্থাৎ অবশেষ ২ । আমার বোঝায় কি কোথাও সমস্যা আছে? Reply শাফায়েত रफद्मग्राति २১, २०১७ at ৯:२१ pm ব্যাপারটা আসলে ঠিক সেরকম না। ধরো একটা সংখ্যা 🗴 । n এর যে মান্টিপলটা 🗴 এর সবথেকে কাছাকাছি পৌছায় কিন্তু 🗴 এর সমান বা ছোট সেই মান্টিপল থেকে 🗴 এর ডিফারেন্সটাই ভাগশেষ। ৫ এর কোন মাল্টিপল -১৭ এর সর্বথেকে কাছাকাছি? অবশ্যই -২০। -১৫ না কারণ এটা -১৭ এর থেকে বড়। -২০ আর -১৭ এর পার্থক্য ৩ তাই ভাগশেষ ৩। Reply **Shikhor Kumer Roy** *(*क्कुःग्राति २১, २०১७ at ১०:১७ pm তাহলে ব্যাপারটা দাড়ালো, যদি বলা হয় -১৭/৫ এর মান কত? উত্তর হবে -৩.৪ [মানে এখানেঃ ভাজ্য, ১৭ = -(৫X৩+২)] আর -১৭/৫ এর ক্ষেত্রে ভাগশেষ কত, তাহলে উত্তর হবে ৩ [মানে এখানেঃ ভাজ্য, ১৭ = ৫X(-৪)+৩] ব্যাপারটা কেমন জানি সাংঘর্ষিক হয়ে গেলো না ভাইয়া? Reply **Mozahidul Islam** ডिসেম্বর ২৫, ২০১৭ at ७:১৩ pm ব্যাপারটা আসলে এই রকম না। আমরা জানি যে ভাজ্য = ভাজক * ভাগফল + ভাগশেষ। আর এখানে এইটাই হয়েছে। -১৭ = ৫ * (-৪) + ৩ । এখানে ভাজ্য = -১৭ , ভাজক = ৫ , ভাগফল = -৪ ও ভাগশেষ = ৩ । আর একটা জিনিস সেটা হচ্ছে ভাগশেষ কখনো ঋণাষ্মক হয় না @Shikhor Kumer Roy ভাই Reply Emtiaj Hasan *रफ*क्रग्राति २२, २०ऽ७ at ১२:७२ am তুমি ফেইসবুকের মাধ্যমে মাসুম বিল্লাল যে কমেন্টটা দিয়েছে সেটা খেয়াল কর। (ভাইয়া আসলে গাণিতিকভাবে -৯%১২=-৯ অথবা ৩। কোনটাই ভুল না। তবে কম্পিউটারের বুদ্ধি নাই যে সে এইটাকে ৩ হিসেবে বলবে 🙂 আসলে ব্যাপারটা হইতেছে -৯ কে এইভাবে দেখেন, ১২ থেকে ৩ সরে আসছে।.) আর দেখ এটা– মনে কর তোমার কাছে এমন একটা ঘড়ি আছে যেটাতে 12 ঘন্টার জায়গায় মাত্র 5 টা ঘন্টা আছে। এদের মার্ক দেয়া 0, 1, 2, 3, 4. জিরো থেকে শুরু করে ঘড়ির কার্টাঁর দিকে ঘুরে তুমি যদি বারো ঘর এগোও তাহলে তুমি 2 এ পৌঁছাবা। গাণিতিকভাবে আসলেই ব্যপারটা হয়- $12 \equiv 2 \pmod{5}$. এখন তুমি যদি ঘড়ির কার্টাঁর বিপরীত দিকে 12 বার ঘুরে আসো তবে তুমি 3 এ এসে পৌ ছাবা। $-12 \equiv 3 \pmod{5}$. @Shikhor Kumar Roy Reply **Nazmul Hasan** ডিসেম্বর ১, ২০১৩ at ২:৩১ am I am grateful to you. When i have not found any effective way to learn algorithm & programming, this is the blog which inspires me to learn algorithm & programming and i enjoy this blog and tutorial every moment. Reply **Jayanto** ডিসেম্বর ৯, ২০১৩ at ১১:88 pm thanks Reply এশা ডিসেম্বর ১৯, ২০১৩ at ৮:২৮ pm খুব ভালো লাগলো ,ভাইয়া । Reply এশা ডিসেম্বর ১৯, ২০১৩ at ৮:৩৫ pm ভাইয়া ,এখানে long long এর জায়গায় i64 ব্যবহার করার কি দরকার ছিল ? Reply শাফায়েত ডিসেম্বর ২৬, ২০১৩ at ১১:৪৩ pm আসলে আমি কোডে সবসময় #define i64 long long এই লাইনটা ব্যবহার করি, i64 আর long long তাই এখানে একই জিনিস। Reply MD. AL-HASAN MRIDHA মার্চ ২৯, ২০১৫ at ১২:৩৩ am ভাইয়া প্রথম প্রব্লেমটাতে wrong answer আসতেছে । বুঝতেছিনা কোথায় ভুল করলাম #include using namespace std; long long int modu(long long int b,long long int p,long long int m) if(p==0)return 1; if(p%2==0)long long int t=modu(b,p/2,m); //return (modu(b,p/2,m)*modu(b,p/2,m))%m;return (t*t)%m; else return ((b%m)*modu(b,p-1,m))%m; int main() long long int b,p,m; cin>>b>>p>>m; cout<<modu(b,p,m);</pre> return 0; Reply প্রান্ত সাহা रफ्कुग्राति २৮, २०১৯ at ৯:०৫ pm **ऽऽ ताषात्र ला**रेत्त , (t*t) %m এই লাইনটা ওভারফ্লো হতে হতে পারে, (t%m * t%m) %m লাইনটা এমন হবে Reply রাইহান হুসাইন আগষ্ট ১৩, ২০১৫ at ১:২২ am ভাইয়া, রিকারসনের কোড টিতে M,N এবং P ভ্যরিয়াবল গুলার কাজ কি আমি বুযিনাই 😟 , দয়াকরে একটু ক্লিয়ার করে বলবেন প্লিজ? 🙂 Reply শাফায়েত আগস্ট २२, २०১৫ at ७:৫२ pm আমরা এখানে (N^P)%M বের করছি। Reply প্রান্তিক সেপ্টেম্বর ১০, ২০১৫ at ৫:২৭ pm ভাইয়া, শেষ কোডটাতে fn() ফাংশনে N parameter হিসাবে নেওয়ার দরকার নাই। Global variable হিসাবে নিলেই চলে। কারণ পুরো fn() ফাংশনে N এর কোনো পরিবর্তন নাই। এটা ঠিক করলে ভালো লাগবে দেখতে। আমার দেখা সবচেয়ে গোছানো সাইট এটা। Thank you for this. Reply শাফায়েত সেপ্টেম্বর ১০, ২০১৫ at ७:৩৫ pm ধন্যবাদ তোমাকে। লোকাল ভ্যারিয়েবল নেয়ার কিছু সুবিধা আছে, ভ্যারিয়েবলের স্কোপটা ওই ফাংশনের মধ্যেই সীমাবদ্ধ থাকে, অন্য কোথাও একই নামের ভ্যারিয়েবল ব্যবহার করতে সমস্যা হয় না। শ্লোবাল ভ্যারিয়েবল বেশি ব্যবহার করলে কোডে বাগ তৈরির সম্ভাবনা বাড়ে :)। Reply প্রান্তিক সেপ্টেম্বর ১১, ২০১৫ at ৯:88 am হুম। বুঝলাম। এখন থেকে Global variable avoid করারার চেষ্টা করব। 🙂 Reply শুভ रफ्कुग्राति ১৯, २०১७ at 8:0১ pm ভাইয়া,চিন্তা করার জন্যে যে প্রবলেমটি দিয়েছেন,সেটা বুঝতেছি না,যদি ক্লিয়ার করতেন,তবে বেশ উপকার হতো! Reply Pingback: শাফায়েতের রূগ » Blog Archive Pingback: শাফায়েতের ৰূগ » Blog Archive কামরান আহমেদ সালমান जुत्तर ७, २०५७ at ১२:৫७ am viya onek upokrito holam . onek donnobad Reply Pingback: Shovon's Note | প্রোগ্রামিং ও এলগরিদম শিখার জন্য সহায়ক রিসোর্স সমূহ Pingback: রবিন-কার্প স্ট্রিং ম্যাচিং | শাফায়েতের ব্লগ Leave a Reply Connect with: Secured by OneAll Social Login Your email address will not be published. Required fields are marked * Comment Name * Email * Website Save my name, email, and website in this browser for the next time I comment. **Post Comment** probhat english phonetic নতুন লেখা ব্যাকএন্ড ইঞ্জিনিয়ারিং: ডিস্ট্রিবিউটেড ফাইল সিস্টেম ব্যাকএন্ড ইঞ্জিনিয়ারিং: কনসিস্টেন্ট হ্যাশিং ব্যাকএন্ড ইঞ্জিনিয়ারিং: লগ-স্ট্রাকচার্ড-ট্রি ডাইনামিক প্রোগ্রামিং ৯ (ট্রি, মিনিমাম ভারটেক্স কভার) ডাইনামিক প্রোগ্রামিং ৮ (বিটমাস্ক, ট্রাভেলিং সেলসম্যান) This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. **Disclaimer:** The advertisement shown in this site is automatically inserted by Google Adsense based on individual user's interest. It doesn't reflect the interest or ideology of the site owner. AccessPress Staple | WordPress Theme: AccessPress Staple by AccessPress Themes