

একটি সংখ্যাকে আসলে যেকোনো সংখ্যা দিয়েই তুমি ভাগ করে দিতে পারে,শূন্য ছাড়া। হ্যা,আমি মিথ্যা বলিনি,চোখ কপালে তোলার মতো কোনো কথাও এটি ছিলো না। অবশ্যই তুমি যেকোনো সংখ্যাকে "যেকোনো সংখ্যা" দিয়ে ভাগ করতে পারবে (শূন্য বাদ -_-) , কিন্তু ব্যাপারটি হচ্ছে, ভাগশেষ সবসময় শূন্য পাবে না! 😊 একটি সংখ্যাকে তুমি আরেকটি সংখ্যা দিয়ে ভাগ করার পর যখন দেখবে কোনো ভাগশেষই অবশিষ্ট থাকছেনা,তখন তুমি সেই সংখ্যাটাকে বড় সংখ্যাটির একটি "বিভাজক" হিসেবে ধরে নিতে পারো। 😊 আজকের এই আলোচনাটি একটি সংখ্যার বিভাজকগুলো নির্ণয় করা নিয়েই! তো শুরু করা যাক।

১) Linear Checking : একটি একটি করে চেক করে বের করাঃ

আমাদের কাজ শুরুর জন্য একটি সংখ্যাকে প্রথমেই ধরে নিই। নিলাম ১২ কে। আচ্ছা,১২ কে কোন কোন সংখ্যা দিয়ে নিঃশেষে ভাগ করা যায় বলতে পারো? চলো একটু পরীক্ষা করে দেখি। ধরে নিলাম যে আমরা কোনকিছুই জানিনা,সিম্পল যোগ বিয়োগ গুণ ভাগ ছাড়া আমাদের তেমন গাণিতিক ধারণা নেই – এটিই ধরে নিলাম এখন। 😊

$$12/1 = 12$$

$$12/2 = 6$$

$$12/3 = 4$$

$$12/4 = 3$$

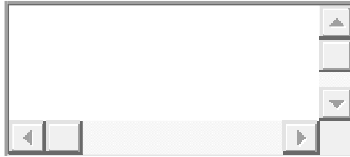
$$12/5 = 2.4 \text{ (উহু,এই ৫ তাহলে বিভাজক না,ভাগফল ভগ্নাংশ চলে আসার মানে তো এর ভাগশেষ আছে!)}$$

$$12/6 = 2$$

এর পর ৭,৮,৯,১০,১১ এদের কেউই বিভাজক হবেনা। একেবারে ১২ কে ১২ দিয়ে ভাগ করলে আমরা পাবো ১ ! তাহলে ১২ এর বিভাজকগুলো আমরা পেয়ে গেলাম,এদেরকে সারিবদ্ধভাবে লিখে ফেলিঃ

১২ এর বিভাজকসমূহ : ১,২,৩,৪,৬,১২

ঠিক এই প্রসেসে যদি আমরা একটি সংখ্যার বিভাজকগুলোকে বের করতে চাই,তাহলে ১ থেকে শুরু করে ঐ সংখ্যাটি পর্যন্ত সবগুলো সংখ্যা দিয়ে ভাগ করে করে চেক করে আমরা বিভাজক বের করতে পারি। এক্ষেত্রে একটি লুপ চালিয়ে টেস্ট করতে করতে বিভাজক হলে প্রিন্ট করে দিলেই কাজ শেষ! কোডটি হবে এরকমঃ (আমি সি/সি++ ব্যবহার করি)



```
1  #include <stdio.h>
2
3  int main()
4  {
5      int number;
6      scanf("%d",&number);
7      int i;
8
9      for(i=1; i<=number; i++)
10     {
11         if(number%i==0) printf("%d ",i);
12     }
13     return 0;
14 }
```

এভাবে খুব সহজেই চাইলে তুমি কোনো একটি সংখ্যার বিভাজকগুলোকে বের করে ফেলতে পারো। কিন্তু প্রোগ্রামারদের চিন্তাভাবনায় হতে হয় স্মার্ট! এমন কোনো প্রক্রিয়া তাদের বেছে নিতে হয়,যেটি সময় সাপেক্ষ। একই কাজ যেটি অনেক গুণ কম সময়ে শেষ করতে পারে। আমরা এখন দেখবো, বিভাজকগুলোকে আরো এফিশিয়েন্ট কোনো ওয়েতে বের করা যায় কিনা! 😊

২) "পরিশ্রম অর্ধেক" প্রসেসঃ

আচ্ছা, আমরা যখন কোনো একটি সংখ্যার বিভাজকগুলো বের করি, তখন কি আমরা খেয়াল করে দেখেছি যে একটি সংখ্যা যদি n হয়, তাহলে সংখ্যাটি ছাড়া সেই সংখ্যাটির কোনো বিভাজকই $n/2$ এর চাইতে বড় হয়না? যেমন ধরো ১২ এর কথাই বলি! এর সবচাইতে বড় বিভাজক ১২ নিজেই, তাই এই ব্যক্তি হিসাবের বাইরে। ঠিক ১২ এর চাইতে ছোট বিভাজকটা কে দেখো তো? ৬ না? 😊 ৬ কি ১২ এর অর্ধেক না? 😊

আরেকটি উদাহরণ নিতে পারো। ধরো, আমরা নিলাম ৩৩। এর বিভাজকগুলো হচ্ছে ১, ৩, ১১, ৩৩। ৩৩ বাদে দ্বিতীয় সর্বোচ্চ বিভাজকটি হচ্ছে ১১। ৩৩ কে অর্ধেক করে দিলে পাই ১৬.৫। তাকিয়ে দেখো, ১১ ১৬.৫ এর চাইতে ছোট-ই আছে! 😊

ব্যাপারটি আসলে কেন ঘটে তা বলি, একটি সংখ্যার "সবচাইতে ছোট" বিভাজকটি হতে পারে ২, কেননা ১ এবং সংখ্যাটি নিজে তো সব সংখ্যার স্কেলট্রেই চলে আসে, তাই তাদেরকে হিসাব থেকে বাদ দিয়ে দিলাম। এখন সংখ্যাটিকে এই ২ দিয়ে ভাগ করে দিলে সংখ্যাটি হয়ে যাচ্ছে অর্ধেক! ব্যাপারটি এমন হতে পারে এখন, যে এই অর্ধেক চেহারার সংখ্যাটা নিজেই একটা বিভাজক, না হয় তার চেয়ে ছোট কেউ। সর্বোচ্চ হলে সংখ্যাটি নিজেই বিভাজক হতে পারে, তার চেয়ে বড় তো আর সম্ভব না, তাই না? 😊 এই ধরনের সংখ্যাজোড় কে আসলে বলে co-factors. ৩ নং পয়েন্টে আমরা এটা ভালোভাবে দেখবো।

তার মানে যা দাঁড়াচ্ছে, তা হচ্ছে আমরা এখন ১ থেকে শুরু করে $n/2$ পর্যন্ত চেক করেই কাজ শেষ করে ফেলতে পারবো। কষ্ট করে n পর্যন্ত চেক করার দরকার পড়ছে না। 😊 কোডটা হবে এমনঃ



```
1  #include <stdio.h>
2
3  int main()
4  {
5      int number;
6      scanf("%d",&number);
7
8      int i;
9
10     for(i=1; i<=number/2; i++)
11     {
12         if(number%i==0) printf("%d ",i);
13     }
14     printf("%d ",number);
15     return 0;
16 }
```

ব্যস, কাজ শেষ! 😊 এর মাধ্যমে ১ নং পদ্ধতির চাইতেও অর্ধেক সময়ে কাজ শেষ করে ফেলা যাবে। যেহেতু আমরা $n/2$ পর্যন্ত চেক করতেছি, তাই সংখ্যাটি নিজে চেকিং থেকে বাদ পড়ে যাচ্ছে। তাই একে আলাদা ভাবে নিচে প্রিন্ট করে দিলাম।

এবার আরেকটু এফিশিয়েন্ট কিভাবে করা যায়, তা নিয়ে ভাবা যাকঃ

৩) "পরিশ্রম স্কয়ার রুট" প্রসেসঃ

এবার আমরা আরেকটু স্মার্ট এপ্রোচ নিবো। নাম্বার থিওরি যা বলে তা হচ্ছে, একটি সংখ্যার বিভাজকগুলো সবসময় জোড়ায় জোড়ায় থাকে। ধরে নিই, একটি বিভাজক হচ্ছে a , যা n কে নিঃশেষে বিভাজ্য করে। এখন আমরা যদি n কে a দ্বারা ভাগ করে দিই, তাহলে আমরা অবশ্যই একটি পূর্ণসংখ্যা b পাবো যেটি নিজেও n এর একটি বিভাজক! ব্যাপারটি গাণিতিক ভাবে এমনঃ

$$n = a * b$$

তাই আমরা যখন একটি বিভাজক পেয়ে যাচ্ছি, তখন কিন্তু একসাথে একটি না, আসলে দুইটি বিভাজকই পেয়ে যাচ্ছি! একটি বিভাজক যখন পাবো a , তখন সেটি দিয়ে n কে ভাগ করলেই আরেকটি বিভাজক b পাওয়া যাচ্ছে। তাই কষ্ট কিন্তু এখানে আরেকটু কমে গেলো, বলতে গেলে আরো অর্ধেক কমে গেলো! 😊

একটি ব্যতিক্রম উদাহরণ দেখাই, ধরি একটি সংখ্যা ৬৪ যার বিভাজকগুলোকে বের করতে হবে। ৬৪ এর বিভাজকগুলো হচ্ছেঃ

```

C:\Users\User\Desktop\Untitled2.exe
64
1 2 4 8 16 32 64
Process returned 0 (0x0)   execution time : 1.307 s
Press any key to continue.

```

মোটমোট ৭ টি বিভাজক। আমরা $n=a*b$ এই পদ্ধতিতে চেক করি একটুঃ

$a = 1, b = 64 \Rightarrow$ তাই $n=1*64 = 64$;

$a = 2, b = 32 \Rightarrow$ তাই $n=2*32 = 64$;

$a = 4, b = 16 \Rightarrow$ তাই $n=4*16 = 64$;

মোট তিনটা কেইসে আমরা ৬ টি বিভাজক পেয়ে গেলাম, কিন্তু বিভাজক তো আছে ৭ টা। একটা বিভাজক আছে, যে a, b এর মতো জোড়া গঠন না করেই একা একা আছে। \therefore তাহলে? আসলে এমন ব্যাপারটা হয় যখন আমাদের সংখ্যাটা হয় একটি পূর্ণবর্গ সংখ্যা! ওই সংখ্যাটার বর্গমূল করলে সেই বিভাজকটি পাওয়া যাবে, মূলতঃ সেই সংখ্যাটিই এরকম জোড়বিহীন একা একা থাকবে। $n=a*b$ এই থিংগরি মতে, এই বিভাজকটির ক্ষেত্রে $a=b$ 😊 তাই জোড়া হিসেবে না, একই সংখ্যা যেহেতু তাই একাই সে থাকবে।

তাই এটি বুঝতেই পারছো, একটি বর্গসংখ্যার বিভাজকসমূহের সংখ্যা সবসময়েই বিজোড় সংখ্যক হবে। 😊 এবং আমরা এখন এইসব আলোচনা কে এক করে নিয়ে একটি নতুন কোড লিখে ফেলবো, এবং ১ থেকে \sqrt{n} পর্যন্ত চেক করেই সব বিভাজক বের করে ফেলবো! 😊 (কেন \sqrt{n} পর্যন্ত চেক করলেই হবে? উপরের প্যারা ভালোভাবে খেয়াল করলেই বুঝবে)



```

1  #include <stdio.h>
2  #include <algorithm>
3  #include <math.h>
4  using namespace std;
5
6  void factors(long long int fact[], long long int n)
7  {
8      long long int i, j = 0;
9      for(i = 1; i*i <= n; i++)
10     {
11         if(n%i == 0)
12         {
13             fact[j++] = i;
14             if(i != sqrt(n)) fact[j++] = n/i;
15         }
16     }
17     sort(fact, fact+j);
18     for(i = 0; i < j; i++) printf("%lld ", fact[i]);
19 }
20 int main()
21 {
22     long long int n, array[1000];
23     scanf("%lld", &n);
24     factors(array, n);

```

```
25     return 0;  
26 }
```

কোডটি লিখলাম C++ ল্যাংগুয়েজে। আর ব্যবহার করেছি একটি array. বিভাজকগুলো সেগুলোতে সেভ করেছি, তারপর এটিকে সর্ট করে তারপর প্রিন্ট করে দিয়েছি। 😊
ভালো থেকে সবাই।