

# ABC 084 解説

writer: namonakiacc

平成 29 年 12 月 30 日

For international readers: English Editorial starts at page 4.

## A: New Year

12 月 30 日の  $M$  時から次の年になるまでは、 $48 - M$  時間です。よって、 $48 - M$  を出力するとよいです。

```
#include <stdio>
int M;
int main()
{
    scanf("%d",&M);
    printf("%d\n",48-M);
}
```

## B: Postal Code

$S$  の  $A+1$  文字目がハイフンであり、それ以外の全ての文字が 0 以上 9 以下である場合に ‘Yes’、そうでない場合に ‘No’ を出力するとよいです。

```
#include <stdio>
int A,B;
char s[12];
int main()
{
    scanf("%d%d",&A,&B);
    scanf("%s",&s);
    for(int i=0;i<A;i++)if(!('0'<=s[i]&&s[i]<='9'))
    {
        puts("No"); return 0;
    }
    if(s[A]!='-')
    {
        puts("No"); return 0;
    }
    for(int i=A+1;i<=A+B;i++)if(!('0'<=s[i]&&s[i]<='9'))
    {
        puts("No"); return 0;
    }
    puts("Yes");
}
```

## C: Special Trains

駅  $j$  に開通式開始  $t$  秒後に到着したとき、駅  $j+1$  に向かう列車に乗れるのが開通式開始何秒後かを考えると、次のようになります。

- ・  $t < S_j$  ならば開通式開始  $S_j$  秒後
- ・ そうでなく、 $t \% F_j = 0$  ならば開通式開始  $t$  秒後
- ・ そうでないなら、開通式開始  $t + F_j - (t \% F_j)$  秒後

これに従い、全ての駅  $i$  から出発する場合に対してシミュレーションしてみると、 $O(N^2)$  となり、解くことができます。

```
#include <stdio>
int N,C[500],S[500],F[500];
int main()
{
    scanf("%d",&N);
    for(int i=0;i<N-1;i++)scanf("%d%d%d",&C[i],&S[i],&F[i]);
    for(int i=0;i<N;i++)
    {
        int t=0;
        for(int j=i;j<N-1;j++)
        {
            if(t<S[j])t=S[j];
            else if(t%F[j]==0);
            else t=t+F[j]-t%F[j];
            t+=C[j];
        }
        printf("%d\n",t);
    }
}
```

## D: 2017-like Number

1 が素数ではないことに注意しつつ、2 以上 100000 以下の数に対して、素数であるか否かを求めます。

これは、エラトステネスの篩を用いて実現できます。

具体的には、以下のような手順をとります。

- ・リストに 2 から 100000 までの整数を入れる。
- ・リストに入っている、まだ見ていない一番小さい数を  $x$  とする。このとき、 $x$  は素数なので、 $x$  の倍数 ( $x$  を除く) をリストから取り除く。

- ・リストに入っている数に、まだ見ていない数がなくなったら操作を終了する。このとき、2 以上 100000 以下の整数のうち、リストに含まれている数は素数であり、含まれていない数は素数でない。

リストに含まれているか否かをフラグにして保管すると、十分な速さで求めることができます。(  $M$  以下の数に対してエラトステネスの篩を適用した時のオーダーは  $O(M \log(\log M))$  となります。 )

これにより、1 以上 100000 以下の奇数に対して、2017 に似た数 か否かが求められるようになりました。

次に、区間内の 2017 に似た数 を高速に求める方法についてです。

これは、累積和を用いることで実現できます。

具体的には、全ての 1 以上 100000 以下の整数  $x$  に対して、1 以上  $x$  以下の数に 2017 に似た数 がいくつあるかを求めておきます。

$c_i = 1$  以上  $x$  以下の 2017 に似た数 の個数 とすると、

- ・  $c_{i+1} = c_i + 1$  ( $i + 1$  が 2017 に似た数 である場合)
- ・  $c_{i+1} = c_i$  ( $i + 1$  が 2017 に似た数 でない場合)

となるので、この部分に関しても十分な速さで求めることができます。

このとき、 $l$  以上  $r$  以下の整数に、2017 に似た数 は  $c_r - c_{l-1}$  個あります。

これらに基づいて実装すると、解くことができます。

```
#include <stdio>
bool f[100001];
int c[100002];
int N,L,R;
int main()
{
    for(int i=2;i<=100000;i++)if(!f[i])for(int j=i+i;j<=100000;j+=i)
        f[j]=true;
    for(int i=3;i<=100000;i+=2)if(!f[i]&&!f[(i+1)/2])c[i]++;
    for(int i=3;i<=100000;i++)c[i]+=c[i-1];
    scanf("%d",&N);
    while(N--)
    {
        scanf("%d%d",&L,&R);
        printf("%d\n",c[R]-c[L-1]);
    }
}
```

## A: New Year

There are  $48 - M$  hours until New Year at  $M$  o'clock on 30th Monday, December.

So you can get 100 point by print  $48 - M$ .

```
#include <stdio>
int M;
int main()
{
    scanf ("%d",&M);
    printf ("%d\n",48-M);
}
```

## B: Postal Code

You can get 200 point by print 'Yes' when  $(A + 1)$  -th character of  $S$  is a hyphen and the other characters are digits from '0' through '9', 'No' otherwise.

```
#include <stdio>
int A,B;
char s[12];
int main()
{
    scanf ("%d%d",&A,&B);
    scanf ("%s",&s);
    for(int i=0;i<A;i++) if (!('0'<=s[i]&& s[i]<='9'))
    {
        puts("No"); return 0;
    }
    if (s[A]!='-')
    {
        puts("No"); return 0;
    }
    for(int i=A+1;i<=A+B;i++) if (!('0'<=s[i]&& s[i]<='9'))
    {
        puts("No"); return 0;
    }
    puts("Yes");
}
```

## C: Special Trains

First, considering when it is possible to ride a train which goes to station  $j + 1$ , in the situation that arriving station  $j$ ,  $t$  seconds after the ceremony begin.

- If  $t < S_j$ ,  $S_j$  seconds after the ceremony begin.
- If  $t \geq S_j$ , but  $t \% F_j = 0$ ,  $t$  seconds after the ceremony begin.
- Otherwise,  $t + F_j - (t \% F_j)$  seconds after the ceremony begin.

Considering this, simulate in every case, it would be  $O(N^2)$  and you can get 300 points.

```
#include <stdio>
int N,C[500],S[500],F[500];
int main()
{
    scanf("%d",&N);
    for(int i=0;i<N-1;i++) scanf("%d%d%d",&C[i],&S[i],&F[i]);
    for(int i=0;i<N;i++)
    {
        int t=0;
        for(int j=i;j<N-1;j++)
        {
            if(t<S[j])t=S[j];
            else if(t%F[j]==0);
            else t=t+F[j]-t%F[j];
            t+=C[j];
        }
        printf("%d\n",t);
    }
}
```

## D: 2017-like Number

Pay attention to the fact that 1 is not prime number, calculate if the number is prime number or not, in the number from 2 to 100000.

It could be done by using Sieve of Eratosthenes.

In more details, do as such.

- put the all number from 2 to 100000 in the list.
- Assume that the minimum number you don't check in the list, as  $x$ .  $x$  is prime number, so remove the multiples of  $x$  (without  $x$ ) from the list.

- If there are no number you don't check in the list, stop this. After that, numbers from 2 to 100000 in the list is prime number, not in the list is not prime number.

Using flag to note whether or not the number is in the list, it could be done by reasonable time. (If you done Sieve of Eratosthenes in the number from 2 to  $M$ ,  $O(M \log(\log M))$ )

Doing so, you can check whether or not  $x$  is similar to 2017, in  $x$  from 1 to 100000.

Next, we should consider about how to solve quickly, how many number is similar to 2017, in some range.

It can be done by Cumulative sum.

In more detail, solve how many number is similar to 2017 in range from 1 to  $x$ , in all number  $x$  from 1 to 100000.

Considering  $c_i$  as showing how many number is similar to 2017 in range from 1 to  $x$ ,

- $c_{i+1} = c_i + 1$  (when  $i + 1$  is similar to 2017)
- $c_{i+1} = c_i$  (Otherwise)

So, it could be done in reasonable time.

And,  $c_r - c_{l-1}$  numbers is similar to 2017 in range from  $l$  to  $r$ .

Doing as such, you can get 400 points.

```
#include <stdio>
bool f[100001];
int c[100002];
int N, L, R;
int main()
{
    for(int i=2; i<=100000; i++) if(!f[i]) for(int j=i+i; j<=100000; j+=i)
        f[j]=true;
    for(int i=3; i<=100000; i+=2) if(!f[i] && !f[(i+1)/2]) c[i]++;
    for(int i=3; i<=100000; i++) c[i] += c[i-1];
    scanf("%d", &N);
    while(N--)
    {
        scanf("%d%d", &L, &R);
        printf("%d\n", c[R] - c[L-1]);
    }
}
```