

Time limit	1 s
Memory limit	64 MB

**Buat program dengan nama kotakbola.c**

Tuan Bil memiliki toko mainan anak-anak berskala multinasional. Sebagai pekerja kelas bawah di perusahaan, anda dipekerjakan untuk memasukan bola-bola mainan berdasarkan warnanya ke kontainer yang sesuai!

Dalam pabrik, kamu memiliki 3 buah kontainer berwarna hijau, merah, dan biru dengan kapasitas masing-masing yaitu kH, kM, dan kB. Bola-bola akan berjalan di belt dengan kode (W) dan jumlah yang berjalan (N). Pengemasan akan berhenti ketika didapatkan kode 'E'. Jika jumlah bola yang diberikan melebihi kapasitas kontainer, maka pengemasan tidak bisa dilakukan. Pengemasan juga tidak dapat dilakukan bila tidak ada bola yang diberikan!

Walaupun anda bekerja di pabrik, anda merupakan lulusan dari Institut Tidak Bekerja! Oleh karena itu, buatlah program yang dapat melakukan pengemasan tersebut!

## Format Masukan

Baris pertama berisikan kH, kM, dan kB yang merupakan kapasitas kontainer.

Baris kedua dan seterusnya akan berisikan sebuah karakter kode W. Kode merupakan warna dari bola-bola yang sedang berjalan sehingga bisa berupa 'H', 'M', dan 'B' yaitu hijau, merah, dan biru. Jika kode berupa warna, maka akan diikuti oleh kuantitas bola yang sedang berjalan.

Baris akan berakhir ketika hanya berisi kode 'E'.

## Format Keluaran

Baris pertama berisikan jumlah bola hijau, merah, dan biru yang didapatkan.

Baris kedua berisikan **Tidak** atau **Ya** berdasarkan kemungkinan bola dikemas. Akhiri baris dengan **newline**

## Contoh 1

Masukkan:

```
10 5 2
H 10
M 2
H 5
M 3
E
```

Keluaran:

```
15 5 0
Tidak
```

Penjelasan: Bola hijau yang masuk berjumlah 15, merah berjumlah 5, dan biru berjumlah 0. Karena jumlah hijau melebihi kapasitas kontainer hijau (10) maka pengemasan tidak mungkin dilakukan.

## Contoh 2

Masukkan:

```
5 5 2
H 2
B 2
H 2
E
```

Keluaran:

```
4 0 2
Ya
```

Penjelasan: Bola hijau yang masuk berjumlah 4, merah berjumlah 0, dan biru berjumlah 2. Karena seluruh kontainer dapat menyimpan jumlah setiap bola, pengemasan dapat dilakukan.

## Contoh 3

Masukkan:

7 3 1

E

Keluaran:

0 0 0

Tidak

Penjelasan: Tidak diberikan bola sehingga jumlah 0 dan tidak bisa dikemas.

C

⬆

⬇

⬆

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.00 sec, 1.66 MB
2	5	Accepted	0.00 sec, 1.49 MB
3	5	Accepted	0.00 sec, 1.66 MB
4	5	Accepted	0.00 sec, 1.64 MB
5	5	Accepted	0.00 sec, 1.60 MB
6	5	Accepted	0.00 sec, 1.64 MB
7	5	Accepted	0.00 sec, 1.59 MB
8	5	Accepted	0.00 sec, 1.66 MB
9	5	Accepted	0.00 sec, 1.49 MB
10	5	Accepted	0.00 sec, 1.66 MB
11	5	Accepted	0.00 sec, 1.62 MB
12	5	Accepted	0.00 sec, 1.66 MB
13	5	Accepted	0.00 sec, 1.70 MB
14	5	Accepted	0.00 sec, 1.49 MB
15	5	Accepted	0.00 sec, 1.56 MB
16	5	Accepted	0.00 sec, 1.66 MB
17	5	Accepted	0.00 sec, 1.71 MB
18	5	Accepted	0.00 sec, 1.51 MB
19	5	Accepted	0.00 sec, 1.70 MB
20	5	Accepted	0.00 sec, 1.62 MB



Time limit	1 s
Memory limit	64 MB

Buat program dengan nama **caesar.c**

Mr. Pines memberikan anda tugas untuk mengimplementasikan Caesar Cipher menggunakan bahasa C.

Caesar Cipher adalah metode enkripsi yang menggantikan setiap huruf dalam suatu kata oleh huruf lain yang memiliki selisih posisi tertentu. Misalnya, bila menggunakan selisih 3, A menjadi D, B menjadi E dan C menjadi F sehingga teks "ABC" berubah menjadi "DEF".

Program yang akan diimplementasikan akan mengenkripsikan suatu kata/kalimat dengan caesar cipher. Selisih dari posisi huruf akan didapatkan dari jumlah karakter kata/kalimat yang akan dienkripsi.

Kata/kalimat yang dienkripsi akan disusun sebagai kalimat yang hanya terdiri dari huruf kapital dan tidak memiliki spasi atau simbol lain selain alfabet. Kalimat yang akan dienkripsi pasti akan diakhiri dengan `.` (titik) dan simbol `.` (titik) tidak termasuk ke dalam kalimat tersebut. Karakter `.` (titik) ini tidak termasuk dalam perhitungan panjang string.

## Format Masukan

Suatu string yang akan dienkripsi, diakhiri dengan `.` (titik)

Kata yang dienkripsi akan memiliki panjang maksimal 20 karakter. Masukan dijamin tidak akan kosong, hanya terdiri dari alfabet, tidak memiliki spasi dan pasti hanya terdiri dari huruf kapital.

HINT : Tidak perlu ada tambahan apa apa di scanfnya ya

## Format Keluaran

Baris string yang sudah dienkripsi dengan *newline*.

## Contoh 1

Masukkan:

ABC.

Keluaran:

DEF

Penjelasan:

- Input memiliki 3 buah karakter sehingga kata akan dienkripsikan dengan caesar cipher dengan selisih posisi sebanyak 3 huruf
- A -> D (Maju 3 huruf A -> B -> C -> D)
- B -> E (Maju 3 huruf B -> C -> D -> E)
- C -> F (Maju 3 huruf C -> D -> E -> F)
- Output akan di print menggunakan newline

## Contoh 2

Masukkan:

PINES.

Keluaran:

UNSJX

Penjelasan:

- Input memiliki 5 buah karakter sehingga kata akan dienkripsikan dengan caesar cipher dengan selisih posisi sebanyak 5 huruf
- P -> U (Maju 3 huruf P -> Q -> R -> S -> T -> U)
- Hal yang sama dilakukan untuk seluruh karakter dalam input
- Output akan di print menggunakan newline

HINT : Setiap huruf akan memiliki nilai integer yang merepresentasikan karakter huruf tersebut nilai integer adalah ASCII. Contohnya huruf A akan memiliki nilai 65, huruf B memiliki nilai 66, dan seterusnya. Dengan bahasa C kalian bisa melakukan mengubah suatu karakter ke karakter lain dengan mengubah ASCII dari karakter tersebut.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)



 [caesar.c](http://caesar.c)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.57 MB
2	10	Accepted	0.00 sec, 1.58 MB
3	10	Accepted	0.00 sec, 1.52 MB
4	10	Accepted	0.00 sec, 1.64 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.58 MB
7	10	Accepted	0.00 sec, 1.64 MB
8	10	Accepted	0.00 sec, 1.58 MB
9	10	Accepted	0.00 sec, 1.56 MB
10	10	Accepted	0.00 sec, 1.46 MB

Time limit	1 s
Memory limit	64 MB

# Makan Pempek

Buat program dengan nama pempek.c

Suatu hari, Diadakan lomba makan pempek di kota Bandung. Terdapat Q buah pertandingan. Pada setiap pertandingan, akan ada 2 orang yang bertarung. Berikut adalah detail Permainannya:

- Pada awal permainan, akan terdapat N buah pempek yang akan dimakan serta D yang menyatakan berapa banyak pempek yang harus dimakan oleh salah satu orang ketika ia mendapat giliran.
- 2 orang tersebut akan makan secara selang-seling. Sebut saja peserta tersebut peserta 0 dan peserta 1.
- Kita dapat asumsikan X sebagai jumlah pempek yang masih ada. Apabila  $X \geq D$ , maka peserta tersebut memakan D buah pempek dan total pempek yang ada akan berkurang D buah biji.
- Peserta yang tidak dapat memakan pempek akan dinyatakan kalah.
- Peserta 0 adalah peserta pertama yang akan makan pempek tersebut.

Kamu diminta untuk membantu mencari tahu siapa pemenang dari setiap pertandingan. Keluarkan nilai 0 apabila peserta 0 menang. Keluarkan nilai 1 apabila peserta 1 menang.

## Format Masukan

Baris pertama terdiri dari bilangan yang menyatakan banyak pertandingan. Q baris selanjutnya terdiri dari dua buah bilangan. N yang menyatakan jumlah pempek yang harus dimakan dan D yang menyatakan jumlah pempek yang harus dimakan pada setiap rondonya.

## Format Keluaran

Terdiri dari Q buah barisan yang berisi jawaban dari setiap pertandingan dan diakhiri dengan newline "\n"

## Batasan

Q ( $1 \leq Q \leq 10^5$ )  
N ( $0 \leq N \leq 18 \cdot 10^{18}$ )  
D ( $1 \leq D \leq 18 \cdot 10^{18}$ )

## Contoh Input 1

Masukkan:

```
4
8 2
9 5
10 20
15 5
```

Keluaran:

```
1
0
1
0
```

## Penjelasan

Pada pertandingan pertama, berikut adalah sekuens berjalannya pertandingan tersebut

- Peserta 0 memakan 2 buah pempek, tersisa 6 pempek
- Peserta 1 memakan 2 buah pempek, tersisa 4 pempek
- Peserta 0 memakan 2 buah pempek, tersisa 2 pempek
- Peserta 1 memakan 2 buah pempek, tersisa 0 pempek
- Peserta 0 tidak bisa memakan pempek lagi, sehingga peserta 1 adalah pemenangnya.

Pada pertandingan kedua, berikut adalah sekuens pertandingan tersebut

- Peserta 0 memakan 5 buah pempek, tersisa 4 pempek
- Peserta 1 tidak bisa memakan pempek lagi karena pempek yang ada bernilai kurang dari D. Sehingga peserta 0 adalah pemenangnya.

Pada pertandingan ketiga, berikut adalah sekuens pertandingan tersebut

- Peserta 0 tidak bisa memakan pempek lagi karena pempek yang ada bernilai kurang dari D. Sehingga peserta 1 adalah pemenangnya.

Pada pertandingan keempat, berikut adalah sekuens pertandingan tersebut

- Peserta 0 memakan 5 buah pempek, tersisa 10 buah pempek
- Peserta 1 memakan 5 buah pempek, tersisa 5 buah pempek
- Peserta 0 memakan 5 buah pempek, tersisa 0 buah pempek
- Peserta 1 tidak bisa memakan pempek lagi, sehingga peserta 0 adalah pemenangnya.



[pempek.c](#)

Score: 90

Blackbox

Score: 90

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.66 MB
2	10	Accepted	0.00 sec, 1.57 MB
3	10	Accepted	0.00 sec, 1.67 MB
4	10	Accepted	0.00 sec, 1.64 MB
5	10	Accepted	0.00 sec, 1.61 MB
6	10	Accepted	0.00 sec, 1.63 MB
7	10	Accepted	0.00 sec, 1.65 MB
8	10	Accepted	0.00 sec, 1.71 MB
9	10	Accepted	0.00 sec, 1.57 MB
10	0	Wrong answer	0.00 sec, 1.55 MB

◀ [Praktikum 2](#)

Jump to...

◀ ▶

[Pra-Praktikum 3](#) ▶