

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)**Objective**

This challenge will help you to learn how to take a character, a string and a sentence as input in C.

To take a single character **ch** as input, you can use scanf("%c", &ch); and printf("%c", ch) writes a character specified by the argument char to stdout:

```
char ch;  
scanf("%c", &ch);  
printf("%c", ch);
```

This piece of code prints the character **ch**.

**Task**

You have to print the character, **ch**.

**Input Format**

Take a character, **ch** as input.

**Output Format**

Print the character, **ch**.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char ch;
5     scanf("%c",&ch);
6     printf("%c",ch);
7     return 0;
8 }
```

	Input	Expected	Got	
✓	C	C	C	✓

Passed all tests! ✓

Print the sum and difference of two float variables rounded to one decimal place on a new line.

### Input Format

The first line contains two integers.

The second line contains two floating point numbers.

### Constraints

- $1 \leq \text{integer variables} \leq 10^4$
- $1 \leq \text{float variables} \leq 10^4$

### Output Format

Print the sum and difference of both integers separated by a space on the first line, and the sum and difference of both float (scaled to **1** decimal place) separated by a space on the second line.

### Sample Input

10 4

4.0 2.0

### Sample Output

14 6

6.0 2.0

### Explanation

When we sum the integers **10** and **4**, we get the integer **14**. When we subtract the second number **4** from the first number **10**, we get **6** as their difference.

When we sum the floating-point numbers **4.0** and **2.0**, we get **6.0**. When we subtract the second number **2.0** from the first number **4.0**, we get **2.0** as their difference.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     float c,d;
6     scanf("%d%d",&a,&b);
7     scanf("%f%f",&c,&d);
8     printf("%d%d",a+b,b-a);
9     printf("/n");
10    printf("%.1f%.1f",c+d,c-d);
11    return 0;
12 }
```

<b>Status</b>	Finished
<b>Started</b>	Thursday, 26 December 2024, 8:09 AM
<b>Completed</b>	Thursday, 26 December 2024, 8:25 AM
<b>Duration</b>	15 mins 57 secs

Question **1**

Correct

Marked out of  
3.00

 [Flag question](#)

Goki recently had a breakup, so he wants to have some more friends in his life. Goki has N people who he can be friends with, so he decides to choose among them according to their skills set  $Y_i (1 \leq i \leq n)$ . He wants atleast X skills in his friends. Help Goki find his friends.

---

**INPUT**

First line contains a single integer X - denoting the minimum skill required to be Goki's friend. Next line contains one integer Y - denoting the skill of the person

.

---

**OUTPUT**

Print if he can be friend with Goki. 'YES' (without quotes) if he can be friends with Goki else 'NO' (without quotes).

---

**CONSTRAINTS**

$1 \leq N \leq 1000000$

$1 \leq X, Y \leq 1000000$

**SAMPLE INPUT 1**

100 110

SAMPLE OUTPUT 1

YES

SAMPLE INPUT 2

100 90

SAMPLE OUTPUT 2

NO

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     scanf("%d%d",&x,&y);
6     if(y>=x)
7     {
8         printf("YES");
9     }
10    else
11    {
12        printf("NO");
13    }
14    return 0;
15 }
```

✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00

 Flag question

Before the outbreak of corona virus to the world, a meeting happened in a room in Wuhan. A person who attended that meeting had COVID-19 and no one in the room knew about it! So everyone started shaking hands with everyone else in the room as a gesture of respect and after meeting unfortunately everyone got infected! Given the fact that any two persons shake hand exactly once, Can you tell the total count of handshakes happened in that meeting? Say no to shakehands. Regularly wash your hands. Stay Safe.

**Input Format**

Read an integer N, the total number of people attended that meeting.

**Output Format**

Print the number of handshakes.

**Constraints**

$0 < N < 106$

**SAMPLE INPUT 1**

1

**SAMPLE OUTPUT**

0

**SAMPLE INPUT 2**

2

## SAMPLE OUTPUT 2

1

Explanation Case 1: The lonely board member shakes no hands, hence 0. Case 2: There are 2 board members, 1 handshake takes place.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     printf("%d",n*(n-1)/2);
7     return 0;
8 }
```

	Input	Expected	Got	
✓	1	0	0	✓
✓	2	1	1	✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

In our school days, all of us have enjoyed the Games period. Raghav loves to play cricket and is Captain of his team. He always wanted to win all cricket matches. But only one last Games period is left in school now. After that he will pass out from school. So, this match is very important to him. He does not want to lose it. So he has done a lot of planning to make sure his teams wins. He is worried about only one opponent - Jatin, who is very good batsman. Raghav has figured out 3 types of bowling techniques, that could be most beneficial for dismissing Jatin. He has given points to each of the 3 techniques. You need to tell him which is the maximum point value, so that Raghav can select best technique. 3 numbers are given in input. Output the maximum of these numbers.

Input:

Three space separated integers.

Output:

Maximum integer value

SAMPLE INPUT

8 6 1

SAMPLE OUTPUT

8

Explanation Out of given numbers, 8 is maximum.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d%d%d",&a,&b,&c);
6     if(a>b&&a>c)
7     {
8         printf("%d",a);
9     }
10    else if(b>a&&b>c)
11    {
12        printf("%d",b);
13    }
```

```
2 | int main()
3 | {
4 |     int a,b,c;
5 |     scanf("%d%d%d",&a,&b,&c);
6 |     if(a>b&&a>c)
7 |     {
8 |         printf("%d",a);
9 |     }
10 |    else if(b>a&&b>c)
11 |    {
12 |        printf("%d",b);
13 |    }
14 |
15 |    else
16 |    {
17 |        printf("%d",c);
18 |    }
19 | }
```

	Input	Expected	Got	
✓	81 26 15	81	81	✓

Passed all tests! ✓

Finish review

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Tuesday, 29 October 2024, 11:44 AM
Duration	55 days 5 hours

Question 1

Correct

Marked out of  
3.00

 Flag question

Write a program to read two integer values and print true if both the numbers end with the same digit, otherwise print false. Example: If 698 and 768 are given, program should print true as they both end with 8. Sample Input 1 25 53 Sample Output 1 false Sample Input 2 27 77  
Sample Output 2 true

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c,d;
5     scanf("%d%d",&a,&b);
6     c=a%10;
7     d=b%10;
8     if(c==d)
9     {
10         printf("true");
11     }
12     else
13     {
14         printf("false");
15     }
16 }
17 }
```

```
13 *
14     printf("false");
15     return 0;
16 }
17
18
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	25 53	false	false	✓
✓	27 77	true	true	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00

 Flag question

### Objective

In this challenge, we're getting started with conditional statements.

### Task

Given an integer,  $n$ , perform the following conditional actions:

- If ***n*** is odd, print Weird
- If ***n*** is even and in the inclusive range of **2** to **5**, print **Not Weird**
- If ***n*** is even and in the inclusive range of **6** to **20**, print **Weird**
- If ***n*** is even and greater than **20**, print **Not Weird**

Complete the stub code provided in your editor to print whether or not ***n*** is weird.

### **Input Format**

A single line containing a positive integer, ***n***.

### **Constraints**

- $1 \leq n \leq 100$

### **Output Format**

Print Weird if the number is weird; otherwise, print Not Weird.

### **Sample Input 0**

Weird

### Sample Input 1

24

### Sample Output 1

Not Weird

### Explanation

*Sample Case 0: **n = 3***

**n** is odd and odd numbers are weird, so we print **Weird**.

*Sample Case 1: **n = 24***

**n > 20** and **n** is even, so it isn't weird. Thus, we print **Not Weird**.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
-
```

```
5     scanf("%d",&n);
6     c=n/2;
7     if(c==1)
8         printf("Weird");
9
10    else
11    {
12        if(c==0&&c>=2&&c<=5)
13            printf("Not Weird");
14
15        else
16        {
17            if(c==0&&c>=6&&c<=20)
18                printf("Weird");
19
20            else
21                printf("Not Weird");
22            return 0;
23        }
24    }
25 }
26
27 }
```

	Input	Expected	Got	
✓	3	Weird	Weird	✓
✓	24	Not Weird	Not Weird	✓

Passed all tests! ✓

## Question 3

Correct

Marked out of  
7.00[Flag question](#)

Three numbers form a Pythagorean triple if the sum of squares of two numbers is equal to the square of the third. For example, 3, 5 and 4 form a Pythagorean triple, since  $3^2 + 4^2 = 25 = 5^2$ . You are given three integers, a, b, and c. They need not be given in increasing order. If they form a Pythagorean triple, then print "yes", otherwise, print "no". Please note that the output message is in small letters. Sample Input 1 3 5 4 Sample Output 1 yes Sample Input 2 5 8 2 Sample Output 2 no

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     scanf("%d%d%d",&a,&b,&c);
6     if(a*a+b*b==c*c || b*b+c*c==a*a || a*a+c*c==b*b)
7     {
8         printf("yes");
9     }
10    else
11    {
12        printf("no");
13        return 0;
14    }
15 }
```

	Input	Expected	Got	
✓	3 5	yes	yes	✓

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Thursday, 7 November 2024, 8:53 AM
<b>Duration</b>	46 days 8 hours

Question **1**

Correct

Marked out of  
3.00

 [Flag question](#)

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

Sample Input 1

3

Sample Output 1

Triangle

Sample Input 2

7

Sample Output 2

Heptagon

Sample Input 3

11

Sample Output 3

The number of sides is not supported.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n==3)
7         printf("Triangle");
8     else if(n==4)
9         printf("Quadrilateral");
10    else
11        printf("Not supported");
12 }
```

```
14     else if(n==5)
15         printf("Pentagon");
16
17     else if(n==6)
18         printf("Hexagon");
19
20     else if(n==7)
21         printf("Heptagon");
22
23     else if(n==8)
24         printf("Octagon");
25
26     else if(n== 9)
27         printf("Nonagon");
28
29     else if(n==10)
30         printf("Decagon");
31
32     else if(n>=11)
33         printf("The number of sides is not supported.");
34
35     return 0;
36
37 }
```

	Input	Expected	Got	
✓	3	Triangle	Triangle	✓

✓	7	Heptagon	Heptagon	✓
✓	11	The number of sides is not supported.	The number of sides is not supported.	✓

Passed all tests! ✓

Question 2

Correct

Marked out of  
5.00

 Flag question

The Chinese zodiac assigns animals to years in a 12-year cycle. One 12-year cycle is shown in the table below. The pattern repeats from there, with 2012 being another year of the Dragon, and 1999 being another year of the Hare.

Year	Animal
2000	Dragon
2001	Snake
2002	Horse
2003	Sheep
2004	Monkey
2005	Rooster
2006	Dog
2007	Pig
2008	Rat
2009	Ox
2010	Tiger
2011	Hare

Write a program that reads a year from the user and displays the animal associated with that year. Your program should work correctly for any year greater than or equal to zero, not just the ones listed in the table.

Sample Input 1

2004

Sample Output 1

Monkey

Sample Input 2

2010

Sample Output 2

Tiger

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
```

```
5     scanf("%d%d",&a,&b);
6     b=(a-2000)%12;
7     if(b==0)
8         printf("Dragon");
9     else if(b==1)
10        printf("Snake");
11    else if(b==2)
12        printf("Horse");
13    else if(b==3)
14        printf("Sheep");
15    else if(b==4)
16        printf("Monkey");
17    else if(b==5)
18        printf("Rooster");
19    else if(b==6)
20        printf("Dog");
21    else if(b==7)
22        printf("Pig");
23    else if(b==8)
24        printf("Rat");
25    else if(b==9)
26        printf("Ox");
27    else if(b==10)
28        printf("Tiger");
29    else if(b==11)
30        printf("Hare");
31    return 0;
32 }
```

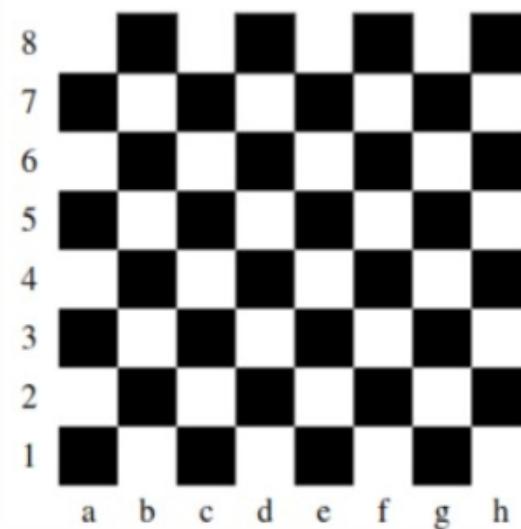
	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:



Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a1 then your program should report that the square is black. If the user enters d5 then your program should report that the square is white. Your program may assume that a valid position will always be entered. It does not need to perform any error checking.

Sample Input 1

a 1

Sample Output 1

The square is black.

Sample Input 1

a 1

Sample Output 1

The square is black.

Sample Input 2

d 5

Sample Output 2

The square is white.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int r;
5     char c;
6     scanf("%c%d",&c,&r);
7     if(c%2==r%2)
8     {
9         printf("The square is black.");
10    }
```

```
4     int r;
5     char c;
6     scanf("%c%d",&c,&r);
7     if(c%2==r%2)
8     {
9         printf("The square is black.");
10    }
11   else
12   {
13       printf("The square is white.");
14   }
15   return 0;
16 }
```

	Input	Expected	Got	
✓	a 1	The square is black.	The square is black.	✓
✓	d 5	The square is white.	The square is white.	✓

Passed all tests! ✓

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Thursday, 21 November 2024, 8:42 AM
<b>Duration</b>	32 days 8 hours

Question **1**

Correct

Marked out of  
3.00

 [Flag question](#)

Some data sets specify dates using the year and day of year rather than the year, month, and day of month. The day of year (DOY) is the sequential day number starting with day 1 on January 1st.

There are two calendars - one for normal years with 365 days, and one for leap years with 366 days. Leap years are divisible by 4. Centuries, like 1900, are not leap years unless they are divisible by 400. So, 2000 was a leap year.

To find the day of year number for a standard date, scan down the Jan column to find the day of month, then scan across to the appropriate month column and read the day of year number. Reverse the process to find the standard date for a given day of year.

Write a program to print the Day of Year of a given date, month and year.

Sample Input 1

18

6

## Sample Output 1

170

**Answer:** (penalty regime: 0 %)

```
7   feb=29;
8   else
9   feb=28;
10  switch(m)
11  {
12      case 1:
13      printf("%d",d);
14      break;
15
16      case 2:
17      printf("%d",31+d);
18      break;
19
20      case 3:
21      printf("%d",31+feb+d);
22      break;
23
24      case 4:
25      printf("%d",31+feb+31+d);
26      break;
27
28      case 5:
29      printf("%d",31+feb+31+30+d);
30      break;
```

```
35
36     case 7:
37     printf("%d",31+feb+31+30+31+30+d);
38     break;
39
40     case 8:
41     printf("%d",31+feb+31+30+31+30+31+d);
42     break;
43
44     case 9:
45     printf("%d",31+feb+31+30+31+30+31+30+d);
46     break;
47
48     case 10:
49     printf("%d",31+feb+31+30+31+30+31+30+31+d);
50     break;
51
52     case 11:
53     printf("%d",31+feb+31+30+31+30+31+30+31+30+d);
54     break;
55
56     case 12:
57     printf("%d",31+feb+31+30+31+30+31+30+31+30+d);
58     break;
59
60 }
61 }
```

	Input	Expected	Got	
✓	18 6 2020	170	170	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of  
5.00

 Flag question

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

- When he says rectangle he is actually referring to a square.
- When he says square, he is actually referring to a triangle.
- When he says triangle he is referring to a rectangle
- And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

**Input Format**

- Name of shape (always in upper case R à Rectangle, S à Square, T à Triangle)
- Length of 1 side
- Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

**Output Format**

- Print the area of the shape.

Sample Input 1

T

10

20

Sample Output 1

200

Sample Input 2

S

30

40

Sample Output 2

600

Sample Input 3

R

10

10

Sample Output 3

100

Sample Input 4

G

8

8

Sample Output 4

0

Sample Input

C

9

10

#### Sample Output 4

0

#### Explanation:

- First is output of area of rectangle
- Then, output of area of triangle
- Then output of area square
- Finally, something random, so we print 0

#### Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     char c;
6     scanf("%c%d%d",&c,&a,&b);
7     switch(c)
8     {
9         case'R':
10            printf("%d",a*b);
11            break;
12         case'C':
```

Question

Correct

Marked out of  
7.00

Flag question

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days: Day Number Name of Day 1 Sunday 2 Monday 3 Tuesday 4 Wednesday 5 Thursday 6 Friday 7 Saturday 8 Kryptoday 9 Coluday 10 Daxamday Here are the rules of the calendar:

- The calendar starts with Sunday always.
- It has only 296 days. After the 296th day, it goes back to Sunday. You begin your journey on a Sunday and will reach after n. You have to tell on which day you will arrive when you reach there.

Input format:

- Contain a number n ( $0 < n$ )

Output format: Print the name of the day you are arriving on

Example Input

7

Example Output

Kryptoday

Example Input

1

Example Output Monday

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n, day;
5     scanf("%d",&n);
6     if(n<296)
7         day=n;
8     else
9         day=n-296;
10    day%=10;
11    day=day+1;
12    day%=10;
13    switch(day){
```

```
29         break;
30
31     case 5:
32     printf("Thursday");
33     break;
34
35     case 6:
36     printf("Friday");
37     break;
38
39     case 7:
40     printf("Saturday");
41     break;
42
43     case 8:
44     printf("Kryptonday");
45     break;
46
47     case 9:
48     printf("Coluday");
49     break;
50
51     case 10:
52     printf("Daxamday");
53     break;
54
55 }
```

	Input	Expected	Got	
✓	7	Kryptonday	Kryptonday	✓
✓	1	Monday	Monday	✓

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Thursday, 21 November 2024, 9:03 AM
<b>Duration</b>	32 days 8 hours

Question **1**

Correct

Marked out of  
3.00

 [Flag question](#)

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

**Input Format**

First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

**Output Format**

Print "Yes" in the case Alice wins, else print "No".

**Constraints**

$1 \leq T \leq 1000$

$1 \leq N \leq 10000$

#### Sample Input and Output

##### Input

3  
1  
6  
7

##### Output

Yes  
Yes  
No

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
```

```
4 |     int i, l=0, n, t;
5 |     scanf("%d", &t);
6 |     while(i < T)
7 |     {
8 |         scanf("%d", &n);
9 |         t = n / 4;
10 |        if(t % 2 == 0 && n % 2 == 0)
11 |        {
12 |            printf("No\n");
13 |        }
14 |        else if(t % 2 == 1 && n % 2 == 1)
15 |        {
16 |            printf("No\n");
17 |        }
18 |        else
19 |        {
20 |            printf("Yes\n");
21 |        }
22 |        i++;
23 |    }
24 |}
```

	Input	Expected	Got	
✓	3	Yes	Yes	✓
	1	Yes	Yes	
	6	No	No	
	7			

Passed all tests! ✓

Correct

Marked out of  
5.00

 Flag question

You are designing a poster which prints out numbers with a unique style applied to each of them. The styling is based on the number of closed paths or holes present in a given number.

The number of holes that each of the digits from 0 to 9 have are equal to the number of closed paths in the digit. Their values are:

1, 2, 3, 5, and 7 = 0 holes.

0, 4, 6, and 9 = 1 hole.

8 = 2 holes.

Given a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has 3 holes.

Complete the program, it must return an integer denoting the total number of holes in num.

#### Constraints

$1 \leq \text{num} \leq 10^9$

#### Input Format For Custom Testing

There is one line of text containing a single integer num, the value to process.

#### Sample Input

Sample Output

2

Explanation

Add the holes count for each digit, 6, 3 and 0. Return  $1 + 0 + 1 = 2$ .

Sample Case 1

Sample Input

1288

Sample Output

4

Explanation

Add the holes count for each digit, 1, 2, 8, 8. Return  $0 + 0 + 2 + 2 = 4$ .

Add the holes count for each digit, 1, 2, 8, 8. Return  $0 + 0 + 2 + 2 = 4$ .

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,n=0;
5     scanf("%d",&a);
6     while(a>0)
7     {
8         b=a%10;
9         if(b==0||b==6||b==9||b==4)
10        {
11            n=n+1;
12        }
13        else if(b==8)
14        {
15            n=n+2;
16        }
17        a=a/10;
18    }
19    printf("%d",n);
20 }
```

	Input	Expected	Got	
✓	630	2	2	✓
✓	1288	4	4	✓

**Question 3**

Correct

Marked out of  
7.00 Flag question

The problem solvers have found a new Island for [coding](#) and named it as Philaland. These smart people were given a task to make a purchase of items at the Island easier by distributing various coins with different values. Manish has come up with a solution that if we make coins category starting from \$1 till the maximum price of the item present on Island, then we can purchase any item easily. He added the following example to prove his point.

Let's suppose the maximum price of an item is 5\$ then we can make coins of {\$1, \$2, \$3, \$4, \$5}to purchase any item ranging from \$1 till \$5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution {\$1, \$2, \$3}. According to him any item can be purchased one time ranging from \$1 to \$5. Everyone was impressed with both of them. Your task is to help Manisha come up with a minimum number of denominations for any arbitrary max price in Philaland.

**Input Format**

Contains an integer N denoting the maximum price of the item present on Philaland.

**Output Format**

Print a single line denoting the minimum number of denominations of coins required.

**Constraints**

$1 \leq T \leq 100$

$1 \leq N \leq 5000$

**Refer the sample output for formatting**

**Sample Input 1:**

10

**Sample Output 1:**

4

**Sample Input 2:**

5

**Sample Output 2:**

3

**Explanation:**

For test case 1, N=10.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,r=0;
5     scanf("%d",&n);
6     while(n!=0)
7     {
8         n=n/2;
9         r=r+1;
10    }
11    printf("%d",r);
12 }
```

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓

**Status** Finished

**Started** Monday, 23 December 2024, 5:33 PM

**Completed** Thursday, 28 November 2024, 9:40 AM

**Duration** 25 days 7 hours

Question **1**

Correct

Marked out of  
3.00

 Flag question

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

Input Format:

The first line will contain the N numbers separated by one space.

Boundary Conditions:

$3 \leq N \leq 50$

The value of the numbers can be from -99999999 to 99999999

Output Format:

The count of numbers where the numbers are odd numbers.

Example Input / Output 1:

Input:

5 10 15 20 25 30 35 40 45 50

Output:

5

Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,x=0;
5     while(scanf("%d",&n)==1)
6     {
7         if(n%2!=0)
8         {
9             x++;
10        }
11    }
12    printf("%d",x);
```

	Input	Expected	Got	
✓	5 10 15 20 25 30 35 40 45 50	5	5	✓

Passed all tests! ✓

## Question 2

Correct

Marked out of

5.00

[Flag question](#)

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively.

When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

### Example 1:

6 -> 9

Input: 6

Output: true

Explanation:

We get 9 after rotating 6, 9 is a valid number and  $9 \neq 6$ .

**Example 2:**

89 -> 68

Input: 89

Output: true

Explanation:

We get 68 after rotating 89, 86 is a valid number and  $86 \neq 89$ .

**Example 3:**

11 -> 11

Input: 11

Output: false

Explanation:

We get 11 after rotating 11, 11 is a valid number but the value remains the same, thus 11 is not a confusing number.

**Note:**

1.  $0 \leq N \leq 10^9$
2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,n,y=1;
5     scanf("%d",&n);
```

```
7 v {  
8   x=n%10;n=n/10;  
9   if(x==2||x==3||x==4||x==5)  
10 v {  
11     y++;  
12   }  
13 }  
14 if(y==1)  
15 v {  
16   printf("true");  
17 }  
18 else  
19 v {  
20   printf("false");  
21 }  
22 }
```

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

Question 3

Correct

Marked out of

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example. food item with value 1 has 1 macronutrient. food item with value 2 has 2 macronutrients. and incrementing in this fashion.

1.00

[Flag question](#)

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given 4 food items (hence value: 1,2,3 and 4), and the unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3 -> the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- $2 + 3 + 4 = 9$
- $1 + 3 + 4 = 8$
- $1 + 2 + 4 = 7$

Since  $2 + 3 + 4 = 9$ , allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo  $1000000007$  ( $10^9 + 7$ ).

It has the following:

- $n$ : an integer that denotes the number of food items
- $k$ : an integer that denotes the unhealthy number

### Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq k \leq 4 \times 10^{15}$

#### Input Format For Custom Testing

The first line contains an integer,  $n$ , that denotes the number of food items.

The second line contains an integer,  $k$ , that denotes the unhealthy number.

#### Sample Input 0

2

2

#### Sample Output 0

3

#### Explanation 0

The following sequence of  $n = 2$  food items:

1. Item 1 has 1 macronutrients.
2.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly  $k = 2$  macronutrients.

**Sample Input 1**

2

1

**Sample Output 1**

2

**Explanation 1**

1. Cannot use item 1 because  $k = 1$  and  $sum \equiv k$  has to be avoided at any time.
2. Hence, max total is achieved by  $sum = 0 + 2 = 2$ .

Sample Case 2

**Sample Input For Custom Testing****Sample Input 2**

3

3

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     long long int n,t,i,nut=0;
5     scanf("%lld%lld",&n,&t);
6     for(i=1;i<=n;i++)
7     {
8         nut=nut+i;
9         if(nut==t)
10        {
11            nut=nut-1;
12        }
13    }
14    printf("%lld",nut%1000000007);
15 }
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3	5	5	✓

Marked out of  
5.00

Flag question

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int rn,n,nt=0,i=0;
5     scanf("%d",&n);
6     do
7     {
8         nt=n;rn=0;
9         while(n!=0)
10        {
11             rn=rn*10+n%10;
12             n=n/10;
13         }
14         n=nt+rn;
15         i++;
16     }
17     while(rn!=nt||i==1);
18
19     printf("%d",rn);
20     return 0;
21 }
22 }
```

	Input	Expected	Got	
✓	32	55	55	✓
✓	789	66066	66066	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of  
7.00

 Flag question

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2.

33344

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n=1,i=0,nt,co=0,e;
5     scanf("%d",&e);
6     while(i<e)
7     {
8         nt=n;
9         while(nt!=0)
10        {
11            co=0;
12            if(nt%10!=3&&nt%10!=4)
13            {
14                co=1;
15                break;
16            }
17            nt=nt/10;
18        }
19        if(co==0)
20        {
21            i++;
22        }
23        n++;
24    }
```

```
8     nt=n;
9     while(nt!=0)
10    {
11        co=0;
12        if(nt%10!=3&&nt%10!=4)
13        {
14            co=1;
15            break;
16        }
17        nt=nt/10;
18    }
19    if(co==0)
20    {
21        i++;
22    }
23    n++;
24 }
25 printf("%d",--n);
26 return 0;
27 }
```

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

Finish review

<b>Status</b>	Finished
<b>Started</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed</b>	Thursday, 19 December 2024, 10:13 AM
<b>Duration</b>	4 days 7 hours

Question **1**

Correct

Marked out of  
5.00

 Flag question

Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their money.

Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.

For example, they have  $m = 6$  to spend and there are flavors costing  $\text{cost} = [1, 2, 3, 4, 5, 6]$ . The two flavors costing **1** and **5** meet the criteria. Using **1**-based indexing, they are at indices **1** and **4**.

### Function Description

Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy.

It has the following:

- $m$ : an integer denoting the amount of money they have to spend

For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

#### Sample Input

```
2
4
5
1 4 5 3 2
4
4
2 2 4 3
```

#### Sample Output

```
1 4
1 2
```

#### Explanation

Sunny and Johnny make the following two trips to the parlor:

1. The first time, they pool together  $m = 4$  dollars. Of the five flavors available that day, flavors **1** and **4** have a total cost of  $1 + 3 = 4$ .
2. The second time, they pool together  $m = 4$  dollars. Of the four flavors available that day, flavors **1** and **2** have a total cost of  $2 + 2 = 4$ .

**Answer:** (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#include<stdio.h>
int main()
{
    int t,m,n,c=0;
    scanf("%d",&t);
    for(int i=0;i<t;i++)
    {
        c=0;
        scanf("%d\n%d",&m, &n);
        int arr[n];
        for(int j=0;j<n;j++)
        {
            scanf("%d",&arr[j]);
        }
        for(int a=0;a<n-1;a++)
        {
            for(int b=a+1;b<n;b++)
            {
```

```
for(int a=0;a<n-1;a++)
{
    for(int b=a+1;b<n;b++)
    {
        if(arr[a]+arr[b]==m)
        {
            printf("%d %d\n",a+1,b+1);
            c=1;
            break;
        }
    }if(c==1)
    break;
}
return 0;
}
```

	Input	Expected	Got	
✓	2 4 5 1 4 5 3 2 4 4 2 2 4 3	1 4 1 2	1 4 1 2	✓

Passed all tests! ✓

**Status** Finished

**Started** Tuesday, 24 December 2024, 11:01 AM

**Completed** Tuesday, 24 December 2024, 12:32 PM

**Duration** 1 hour 30 mins

**Question 1**

Correct

Marked out of  
1.00

 Flag question

You are given a two-dimensional 3\*3 array starting from A [0][0]. You should add the alternate elements of the array and print its sum. It should print two different numbers the first being sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2 and A 0 1, A 1 0, A 1 2, A 2 1.

**Input Format**

First and only line contains the value of array separated by single space.

A 0 0	A 0 1	A 0 2
4	6	9
A 1 0	A 1 1	A 1 2
2	5	8
A 2 0	A 2 1	A 2 2
1	3	7

**Status** Finished

**Started** Tuesday, 24 December 2024, 10:22 AM

**Completed** Tuesday, 24 December 2024, 11:00 AM

**Duration** 37 mins 18 secs

Question **1**

Correct

Marked out of  
1.00

 Flag question

Coders here is a simple task for you, you have given an array of size **N** and an integer **M**.

Your task is to calculate the ***difference between maximum sum and minimum sum of N-M elements*** of the given array.

**Constraints:**

***1<=t<=10***

***1<=n<=1000***

***1<=a[i]<=1000***

**Input:**

First line contains an integer **T** denoting the number of testcases.

First line of every testcase contains two integer **N** and **M**.

Next line contains **N** space separated integers denoting the elements of array

**Output:**

For every test case print your answer in new line

SAMPLE INPUT

```
1
5 1
1 2 3 4 5
```

SAMPLE OUTPUT

```
4
```

Explanation

M is 1 and N is 5 so you have to calculate maximum and minimum sum using (5-1 =) 4 elements.

Maximum sum using the 4 elements would be  $(2+3+4+5=)14$ .

Minimum sum using the 4 elements would be  $(1+2+3+4=)10$ .

Difference will be  $14-10=4$ .

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int t;
5     scanf("%d",&t);
6     while(t--)
7     {
8         int n,m,d,min,temp;
9         scanf("%d%d",&n,&m);
10        d=n-m;
11        int arr[n];
12        for(int i=0;i<n;i++)
13            scanf("%d",&arr[i]);
14        for(int j=0;j<n;j++)
15        {
16            min=j;
17            for(int k=j;k<n;k++)
18            {
19                if(arr[k]<arr[min])
20                    min=k;
21            }
22            temp=arr[min];
23            arr[min]=arr[j];
24            arr[j]=temp;
25        }
26        int maxsum=0,minsum=0;
27        for(int a=0;a<d;a++)
28            minsum+=arr[a];
29        for(int b=n-1;b>m-1;b--)
30            maxsum+=arr[b];
31        printf("%d\n",maxsum-minsum);
32    }
33 }
```

	Input	Expected	Got	
✓	1 5 1 1 2 3 4 5	4	4	✓

Passed all tests! ✓

### Question 2

Correct

Marked out of  
1.00

 Flag question

A new deadly virus has infected large population of a planet. A brilliant scientist has discovered a new strain of virus which can cure this disease. Vaccine produced from this virus has various strength depending on midichlorians count. A person is cured only if midichlorians count in vaccine batch is more than midichlorians count of person. A doctor receives a new set of report which contains midichlorians count of each infected patient. Practo stores all vaccine doctor has and their midichlorians count. You need to determine if doctor can save all patients with the vaccines he has. The number of vaccines and patients are equal.

#### Input Format

First line contains the number of vaccines - N. Second line contains N integers, which are strength of vaccines. Third line contains N integers, which are midichlorians count of patients.

#### Output Format

Print a single line containing '**Yes**' or '**No**'.

#### Input Constraint

**1 < N < 10**

Strength of vaccines and midichlorians count of patients fit in integer.

**SAMPLE INPUT**

```
5
123 146 454 542 456
100 328 248 689 200
```

**SAMPLE OUTPUT**

No

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,min1,min2,temp,flag=0;
5     scanf("%d",&n);
6     int vac[n],pat[n];
7     for(int i=0;i<n;i++)
8         scanf("%d",&vac[i]);
```

	Input	Expected	Got	
✓	5 123 146 454 542 456 100 328 248 689 200	No	No	✓

Passed all tests! ✓

Question 3

Correct

Marked out of  
1.00

 Flag question

You are given an array of  $n$  integer numbers  $a_1, a_2, \dots, a_n$ . Calculate the number of pair of indices  $(i, j)$  such that  $1 \leq i < j \leq n$  and  $a_i \text{ xor } a_j = 0$ .

#### Input format

- First line:  $n$  denoting the number of array elements
- Second line:  $n$  space separated integers  $a_1, a_2, \dots, a_n$ .

#### Output format

Output the required number of pairs.

#### Constraints

$$1 \leq n \leq 10^6$$

$$1 \leq a_i \leq 10^9$$

### SAMPLE INPUT

```
5  
1 3 1 4 3
```

### SAMPLE OUTPUT

```
2
```

#### Explanation

The 2 pair of indices are **(1, 3)** and **(2,5)**.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int n,count=0;  
5     scanf("%d",&n);  
6     int arr[n];  
7     for(int i=0;i<n;i++)  
8         scanf("%d",&arr[i]);  
9     for(int i=0;i<n-1;i++)
```

```
8     scanf("%d",&arr[i]);
9
10    for(int i=0;i<n-1;i++)
11    {
12        for(int j=i+1;j<n;j++)
13        {
14            if((arr[i]^arr[j])==0)
15            count++;
16        }
17    }
18 }
```

	Input	Expected	Got	
✓	5 1 3 1 4 3	2	2	✓

Passed all tests! ✓

Question 4

Correct

Marked out of  
1.00

Flag question

You are given an array **A** of non-negative integers of size **m**. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

**Example:**

A={4,5,3,7,1}

After sorting the new array becomes A={1,3,4,5,7}.

The required output should be "4 2 0 1 3"

**INPUT :**

The first line of input consists of the size of the array

The next line consists of the array of size m

**OUTPUT :**

Output consists of a single line of integers

**CONSTRAINTS:**

**1<=m<=106**

**0<=A[i]<=106**

NOTE: The indexing of the array starts with 0.

### SAMPLE INPUT

```
5  
4 5 3 7 1
```

### SAMPLE OUTPUT

```
4 2 0 1 3
```

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int n;  
5     scanf("%d",&n);  
6     int arr[n];  
7     for(int i=0;i<n;i++)  
8         scanf("%d",&arr[i]);  
9     int max=arr[0];  
10    for(int i=1;i<n;i++)  
11    {  
12        if(arr[i]>max)  
13            max=arr[i];  
14    }  
15    max++;  
16    int min=0;  
17    for(int a=0;a<n;a++)  
18    {  
19        for(int b=0;b<n;b++)  
20        {
```

```
8     scanf("%d",&arr[i]);
9     for(int i=0;i<n-1;i++)
10    {
11        for(int j=i+1;j<n;j++)
12        {
13            if((arr[i]^arr[j])==0)
14                count++;
15        }
16    }
17    printf("%d",count);
18 }
```

	Input	Expected	Got	
✓	5 1 3 1 4 3	2	2	✓

Passed all tests! ✓

Question 4

Correct

Marked out of  
1.00

Flag question

You are given an array **A** of non-negative integers of size **m**. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

**Example:**

$A = \{4, 5, 3, 7, 1\}$

After sorting the new array becomes  $A = \{1, 3, 4, 5, 7\}$ .

The required output should be "4 2 0 1 3"

**INPUT :**

The first line of input consists of the size of the array

The next line consists of the array of size m

**OUTPUT :**

Output consists of a single line of integers

**CONSTRAINTS:**

$1 \leq m \leq 10^6$

$0 \leq A[i] \leq 10^6$

NOTE: The indexing of the array starts with 0.

### SAMPLE INPUT

```
5  
4 5 3 7 1
```

### SAMPLE OUTPUT

```
4 2 0 1 3
```

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int n;  
5     scanf("%d",&n);  
6     int arr[n];  
7     for(int i=0;i<n;i++)  
8         scanf("%d",&arr[i]);  
9     int max=arr[0];  
10    for(int i=1;i<n;i++)  
11    {  
12        if(arr[i]>max)  
13            max=arr[i];  
14    }  
15    max++;  
16    int min=0;  
17    for(int a=0;a<n;a++)  
18    {  
19        for(int b=0;b<n;b++)  
20        {
```

```
9     int max=arr[0];
10    for(int i=1;i<n;i++)
11    {
12        if(arr[i]>max)
13            max=arr[i];
14    }
15    max++;
16    int min=0;
17    for(int a=0;a<n;a++)
18    {
19        for(int b=0;b<n;b++)
20        {
21            if(arr[b]<arr[min])
22                min=b;
23        }
24        printf("%d ",min);
25        arr[min]=max;
26    }
27 }
```

	Input	Expected	Got	
✓	5 4 5 3 7 1	4 2 0 1 3	4 2 0 1 3	✓

Passed all tests! ✓

### Output Format

First line should print sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2

Second line should print sum of A 0 1, A 1 0, A 1 2, A 2 1

### SAMPLE INPUT

1 2 3 4 5 6 7 8 9

### SAMPLE OUTPUT

25

20

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int arr[3][3];
5     for(int i=0;i<3;i++)
6     {
7         for(int j=0;j<3;j++)
8         {
9             scanf("%d",&arr[i][j]);
10        }
11    }
12    int odd=0,even=0;
```

```

3 *
4 {
5     int arr[3][3];
6     for(int i=0;i<3;i++)
7     {
8         for(int j=0;j<3;j++)
9         {
10            scanf("%d",&arr[i][j]);
11        }
12    int odd=0,even=0;
13    for(int i=0;i<3;i++)
14    {
15        for(int j=0;j<3;j++)
16        {
17            if((i+j)%2!=0)
18                odd+=arr[i][j];
19            else
20                even+=arr[i][j];
21        }
22    }
23    printf("%d\n%d",even,odd);
24 }

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1 2 3 4 5 6 7 8 9	25 20	25 20	✓
✓	21 422 423 443 586 645 657 846 904	2591 2356	2591 2356	✓

Passed all tests! ✓

<b>Status</b>	Finished
<b>Started</b>	Sunday, 12 January 2025, 9:23 PM
<b>Completed</b>	Sunday, 12 January 2025, 9:58 PM
<b>Duration</b>	34 mins 49 secs

Question 1

Correct

Marked out of  
1.00

 Flag question

Given a string, **s**, consisting of alphabets and digits, find the frequency of each digit in the given string.

**Input Format**

The first line contains a string, **num** which is the given number.

**Constraints**

$$1 \leq \text{len}(\text{num}) \leq 1000$$

All the elements of num are made of English alphabets and digits.

**Output Format**

Print ten space-separated integers in a single line denoting the frequency of each digit from **0** to **9**.

#### Sample Input 0

a11472o5t6

#### Sample Output 0

0 2 1 0 1 1 1 1 0 0

#### Explanation 0

In the given string:

- **1** occurs two times.
- **2, 4, 5, 6** and **7** occur one time each.

The remaining digits **0, 3, 8** and **9** don't occur at all.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char str[1000];
5     scanf("%s",str);
6     int harsh [10]={0,0,0,0,0,0,0,0,0,0};
7     int temp;
```

```
2 int main()
3 {
4     char str[1000];
5     scanf("%s",str);
6     int harsh [10]={0,0,0,0,0,0,0,0,0,0};
7     int temp;
8     for(int i=0;str[i]!='\0';i++)
9     {
10        temp=str[i]-'0';
11        if(temp<=9&&temp>=0)
12        {
13            harsh[temp]++;
14        }
15    }
16    for(int i=0;i<=9;i++)
17    {
18        printf("%d ",harsh[i]);
19    }
20    return 0;
21 }
```

	Input	Expected	Got	
✓	a11472o5t6	0 2 1 0 1 1 1 1 0 0	0 2 1 0 1 1 1 1 0 0	✓
✓	1w4n88j12n1	0 2 1 0 1 0 0 0 2 0	0 2 1 0 1 0 0 0 2 0	✓
✓	1v88886l256338ar0ekk	1 1 1 2 0 1 2 0 5 0	1 1 1 2 0 1 2 0 5 0	✓

Passed all tests! ✓

Question 2

Correct

Marked out of  
1.00

 Flag question

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

**Note:** The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

**Input:**

The first line consists of an integer  $T$  denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

**Output:**

For each test case, print the count in a new line.

**Constraints:**

$1 \leq T \leq 10$

$1 \leq \text{length of string} \leq 10^5$

**SAMPLE INPUT**

2

nBBZLaosnm

JHklsnZtTL

## SAMPLE OUTPUT

2

1

## Explanation

In test case 1, a and o are the only vowels. So, count=2

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int t;
5     scanf("%d",&t);
6     while(t--)
7     {
8         char str[100000];
9         int count=0;
10        scanf("%s",str);
11        for(int i=0;str[i]!='\0';i++)
12        {
13            char c=str[i];
14            if((c=='a')||(c=='e')||(c=='i')||(c=='o')||(c=='u')||(c=='A')||(c=='E')||(c=='I')||(c=='O')||(c=='U'))
15            count++;
16        }
17    }
18 }
```

```
1 #include<stdio.h>
2 int main()
3 {
4     int t;
5     scanf("%d",&t);
6     while(t--)
7     {
8         char str[100000];
9         int count=0;
10        scanf("%s",str);
11        for(int i=0;str[i]!='\0';i++)
12        {
13            char c=str[i];
14            if((c=='a')||(c=='e')||(c=='i')||(c=='o')||(c=='u')||(c=='A')||(c=='E')||(c=='I')||(c=='O')||(c=='U'))
15            count++;
16        }
17        printf("%d\n",count);
18    }
19    return 0;
20 }
```

	Input	Expected	Got	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓

Passed all tests! ✓

Question 3

Correct

Marked out of  
1.00

 Flag question

Given a sentence,  $s$ , print each word of the sentence in a new line.

#### Input Format

The first and only line contains a sentence,  $s$ .

#### Constraints

$1 \leq \text{len}(s) \leq 1000$

#### Output Format

Print each word of the sentence in a new line.

#### Sample Input 0

This is C

#### Sample Output 0

This

is

C

## Explanation 0

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char s[1000];
5     scanf("%[^\\n]s",s);
6     for(int i=0;s[i]!='\\0';i++)
7     {
8         if(s[i]!=' ')
9             printf("%c",s[i]);
10        else
11            printf("\\n");
12    }
13    return 0;
14 }
```

	Input	Expected	Got	
✓	This is C	This is C	This is C	✓
✓	Learning C is fun	Learning C is fun	Learning C is fun	✓

Passed all tests! ✓

#### Question 4

Correct

Marked out of  
1.00

 Flag question

#### Input Format

You are given two strings, **a** and **b**, separated by a new line. Each string will consist of lower case Latin characters ('a'-'z').

#### Output Format

In the first line print two space-separated integers, representing the length of **a** and **b** respectively.

In the second line print the string produced by concatenating **a** and **b** (**a + b**).

In the third line print two strings separated by a space, **a'** and **b'**. **a'** and **b'** are the same as **a** and **b**, respectively, except that their first characters are swapped.

#### Sample Input

Duration 40 mins 53 secs

Question 1

Incorrect

Marked out of  
1.00

[Flag question](#)

Two strings **A** and **B** comprising of lower case English letters are compatible if they are equal or can be made equal by following this step any number of times:

- Select a prefix from the string **A** (possibly empty), and increase the alphabetical value of all the characters in the prefix by the same valid amount. For example, if the string is **xyz** and we select the prefix **xy** then we can convert it to **yx** by increasing the alphabetical value by 1. But if we select the prefix **xyz** then we cannot increase the alphabetical value.

Your task is to determine if given strings **A** and **B** are compatible.

**Input format**

First line: String **A**

Next line: String **B**

**Output format**

For each test case, print **YES** if string **A** can be converted to string **B**, otherwise print **NO**.

Constraints

**1 ≤ len(A) ≤ 1000000**

**1 ≤ len(B) ≤ 1000000**

### SAMPLE INPUT

abaca

cdbda

### SAMPLE OUTPUT

YES

### Explanation

The string **abaca** can be converted to **bcbda** in one move and to **cdbda** in the next move.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str1[1000000],str2[1000000];
6     int flag=1;
7     scanf("%s",str1);
8     scanf("%s",str2);
9     int a=strlen(str1);
10    int b=strlen(str2);
11    if(a==b)
12    {
```

Marked out of  
1.00

 Flag question

by him did not taste good :(. Joey is feeling extremely hungry and wants to eat pizza. But he is confused about the restaurant from where he should order. As always he asks Chandler for help.

Chandler suggests that Joey should give each restaurant some points, and then choose the restaurant having **maximum points**. If more than one restaurant has same points, Joey can choose the one with **lexicographically smallest** name.

Joey has assigned points to all the restaurants, but can't figure out which restaurant satisfies Chandler's criteria. Can you help him out?

**Input:**

First line has N, the total number of restaurants.

Next N lines contain Name of Restaurant and Points awarded by Joey, separated by a space. Restaurant name has **no spaces**, all lowercase letters and will not be more than 20 characters.

**Output:**

Print the name of the restaurant that Joey should choose.

**Constraints:**

$$1 \leq N \leq 10^5$$

$$1 \leq \text{Points} \leq 10^6$$

**SAMPLE INPUT**

3

Pizzeria 108

Dominos 145

Pizzapizza 49

## SAMPLE OUTPUT

Dominos

## Explanation

**Dominos** has maximum points.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     int n;
6     scanf("%d",&n);
7     char res[n][21];
8     int rate[n];
9     for(int i=0;i<n;i++)
10    {
11        scanf("%s",res[i]);
12        scanf("%d",&rate[i]);
13    }
```

```

15
16     char ans[20];
17     strcpy(ans,res[0]);
18     for(int i=1;i<n;i++)
19     {
20         if(rate[i]>max)
21         {
22             max=rate[i];
23             strcpy(ans,res[i]);
24         }
25         else if(rate[i]==max)
26         {
27             if(strcmp(res[i],ans)<0)
28                 strcpy(ans,res[i]);
29         }
30     printf("%s",ans);
31     return 0;
32 }

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	3 Pizzeria 108 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

Question 4  
Correct

These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are

Marked out of  
1.00

 Flag question

Mobile number can have special characters like '-' and '+' but it can't have prefix zeroes. You have to determine the valid numbers.

You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10 , consists of numeric values and it shouldn't have prefix zeroes.

**Input:**

First line of input is T representing total number of test cases.

Next T line each representing "S" as described in in problem statement.

**Output:**

Print "YES" if it is valid mobile number else print "NO".

Note: Quotes are for clarity.

**Constraints:**

$1 \leq T \leq 10^3$

sum of string length  $\leq 10^5$

**SAMPLE INPUT**

123456789

0123456789

0123456.87

## SAMPLE OUTPUT

YES

NO

NO

**Answer:** (penalty regime: 0 %)

```
21 }  
22     if(s[i]<'0'||s[i]>'9')  
23     {  
24         flag=0;  
25         break;  
26     }  
27 }  
28 else  
29 flag=0;  
30 if(flag==1)  
31 printf("YES\n");  
32 else  
33 printf("NO\n");  
34 }  
35 return 0;  
36 }  
37 }
```

	Input	Expected	Got	
✓	3	YES	YES	✓
	1234567890	NO	NO	
	0123456789	NO	NO	
	0123456.87			

Passed all tests! ✓

Finish review

**Question 1**

Correct

Marked out of  
1.00[Flag question](#)

A binary number is a combination of 1s and 0s. Its  $n^{\text{th}}$  least significant digit is the  $n^{\text{th}}$  digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the  $4^{\text{th}}$  least significant digit.

**Example**

number = 23

- Convert the decimal number 23 to binary number:  $23^{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$ .
- The value of the  $4^{\text{th}}$  index from the right in the binary representation is 0.

**Function Description**

Complete the function fourthBit in the editor below.

fourthBit has the following parameter(s):

int number: a decimal integer

Returns:

int: an integer 0 or 1 matching the 4th least significant digit in the binary representation of number.

**Constraints**

$0 \leq \text{number} < 2^{31}$

### **Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function.

The only line contains an integer, number.

### **Sample Case 0**

#### **Sample Input 0**

STDIN Function

-----

32 → number = 32

#### **Sample Output 0**

0

#### **Explanation 0**

### Sample Input 1

STDIN Function

-----

77 → number = 77

### Sample Output 1

1

### Explanation 1

- Convert the decimal number 77 to binary number:  $77_{10} = (1001101)_2$ .
- The value of the 4th index from the right in the binary representation is 1.

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 /*  
2  * Complete the 'fourthBit' function below.  
3  *  
4  * The function is expected to return an INTEGER.  
5  * The function accepts INTEGER number as parameter.  
6  */  
7 
```

```
3 *
4 * The function is expected to return an INTEGER.
5 * The function accepts INTEGER number as parameter.
6 */
7
8 int fourthBit(int number)
9 {
10    int binary[32];
11    int i=0;
12    while (number>0)
13    {
14
15        |   |   binary[i]=number%2;
16        |   |   number/=2;
17        |   |   i++;
18
19    }
20    if(i>=4)
21    {
22        |   return binary[3];
23    }
24    else
25    return 0;
26 }
```

	Test	Expected	Got	
✓	printf("%d", fourthBit(32))	0	0	✓
✓	printf("%d", fourthBit(77))	1	1	✓

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
1.00[Flag question](#)

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the  $p^{\text{th}}$  element of the list, sorted ascending. If there is no  $p^{\text{th}}$  element, return 0.

**Example**

$n = 20$

$p = 3$

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if  $p = 3$ , then 4 is returned. If  $p > 6$ , 0 would be returned.

**Function Description**

Complete the function `pthFactor` in the editor below.

`pthFactor` has the following parameter(s):

`int n`: the integer whose factors are to be found

`int p`: the index of the factor to be returned

Returns:

`int`: the long integer value of the  $p^{\text{th}}$  integer factor of `n` or, if there is no factor at that index, then 0 is returned

**Constraints**

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^9$

#### Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer  $n$ , the number to factor.

The second line contains an integer  $p$ , the 1-based index of the factor to return.

#### Sample Case 0

#### Sample Input 0

STDIN      Function

-----

10       $\rightarrow n = 10$

3       $\rightarrow p = 3$

#### Sample Output 0

### **Explanation 0**

Factoring  $n = 10$  results in  $\{1, 2, 5, 10\}$ . Return the  $p = 3^{\text{rd}}$  factor, 5, as the answer.

### **Sample Case 1**

#### **Sample Input 1**

STDIN      Function

-----

10       $\rightarrow n = 10$

5       $\rightarrow p = 5$

#### **Sample Output 1**

0

### **Explanation 1**

Factoring  $n = 10$  results in  $\{1, 2, 5, 10\}$ . There are only 4 factors and  $p = 5$ , therefore 0 is returned as the answer.

### **Sample Case 2**

#### **Sample Input 2**

STDIN    Function

-----

1 → n = 1

1 → p = 1

### Sample Output 2

1

### Explanation 2

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 /*  
2  * Complete the 'pthFactor' function below.  
3  *  
4  * The function is expected to return a LONG_INTEGER.  
5  * The function accepts following parameters:  
6  * 1. LONG_INTEGER n  
7  * 2. LONG_INTEGER p  
8  */  
9  
10 long pthFactor(long n, long p)  
11 {  
12     int count=0;
```

```
10 long pthFactor(long n, long p)
11 {
12     int count=0;
13     for (long i=1;i<=n;++i)
14     {
15         if(n%i==0)
16         {
17             count++;
18             if(count==p)
19             {
20                 return i;
21             }
22         }
23     }
24     return 0;
25 }
```

	Test	Expected	Got	
✓	printf("%ld", pthFactor(10, 3))	5	5	✓
✓	printf("%ld", pthFactor(10, 5))	0	0	✓
✓	printf("%ld", pthFactor(1, 1))	1	1	✓

Passed all tests! ✓

Finish review

Duration 11 mins 48 secs

Question 1

Correct

 Flag question

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

### Example

arr=[1,2,3,4,6]

- the sum of the first three elements,  $1+2+3=6$ . The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

### Function Description

Complete the function balancedSum in the editor below.

balancedSum has the following parameter(s):

int arr[n]: an array of integers

Returns:

int: an integer representing the index of the pivot

#### Constraints

- $3 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$ , where  $0 \leq i < n$
- It is guaranteed that a solution always exists.

#### Input Format for Custom Testing

Input from `stdin` will be processed as follows and passed to the function.

The first line contains an integer  $n$ , the size of the array  $arr$ .

Each of the next  $n$  lines contains an integer,  $arr[i]$ , where  $0 \leq i < n$ .

#### Sample Case 0

#### Sample Input 0

#### STDIN    Function Parameters

-----

4 →  $arr[]$  size  $n = 4$

1 →  $arr = [1, 2, 3, 3]$

2

3

3

Sample Output 0

2

Explanation 0

- The sum of the first two elements,  $1+2=3$ . The value of the last element is 3.
- Using zero based indexing,  $\text{arr}[2]=3$  is the pivot between the two subarrays.
- The index of the pivot is 2.

Sample Case 1

Sample Input 1

STDIN    Function Parameters

-----

3    → arr[] size n = 3

1    → arr = [1, 2, 1]

2

1

Sample Output 1

1

**Answer:** (penalty regime: 0 %)

Reset answer

```
1  /*
2  * Complete the 'balancedSum' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER_ARRAY arr as parameter.
6  */
7
8 int balancedSum(int arr_count, int* arr)
9 {
10     int left_sum=0;
11     for(int i=0;i<arr_count;i++){
12         left_sum+=arr[i];
13         for(int j=arr_count;j>arr_count/2;j--){
14             if(left_sum==arr[j]) return arr_count -(i+1);
15         }
16     }
17     return 0;
18 }
19
```

	Test	Expected	Got	
✓	int arr[] = {1,2,3,3}; printf("%d", balancedSum(4, arr))	2	2	✓

Passed all tests! ✓

Question **2**

Correct

 [Flag question](#)

Calculate the sum of an array of integers.

Example

```
numbers = [3, 13, 4, 11, 9]
```

The sum is  $3 + 13 + 4 + 11 + 9 = 40$ .

Function Description

Complete the function `arraySum` in the editor below.

`arraySum` has the following parameter(s):

`int numbers[n]:` an array of integers

Returns

`int:` integer sum of the numbers array

Constraints

$1 \leq n \leq 10^4$

## Explanation 0

$1 + 2 + 3 + 4 + 5 = 15.$

## Sample Case 1

### Sample Input 1

STDIN      Function

-----

2      →    numbers[] size n = 2

12     →    numbers = [12, 12]

12

### Sample Output 1

24

## Explanation 1

$12 + 12 = 24.$

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 /*
2  * Complete the 'arraySum' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER_ARRAY numbers as parameter.
6  */
7
8 int arraySum(int numbers_count, int *numbers)
9 {
10     int sum=0;
11     for(int i=0;i<numbers_count;i++){
12         sum+=numbers[i];
13     }
14     return sum;
15 }
```

	Test	Expected	Got	
✓	int arr[] = {1,2,3,4,5}; printf("%d", arraySum(5, arr))	15	15	✓

Passed all tests! ✓

Question 3

Correct

 Flag question

Given an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences. Example n = 5 arr = [1, 3, 3, 2, 4] If the list is rearranged as arr' = [1, 2, 3, 3, 4], the absolute differences are  $|1 - 2| = 1$ ,  $|2 - 3| = 1$ ,  $|3 - 3| = 0$ ,  $|3 - 4| = 1$ . The sum of those differences is  $1 + 1 + 0 + 1 = 3$ . Function Description Complete the function minDiff in the editor below. minDiff has the following parameter: arr: an integer array Returns: int: the sum of the absolute differences of adjacent elements Constraints  $2 \leq n \leq 105$   $0 \leq arr[i] \leq 109$ , where  $0 \leq i < n$  Input Format For Custom Testing The first line of input contains an integer, n, the size of arr. Each of the following n lines contains an integer that describes arr[i] (where  $0 \leq i < n$ ) . Sample Case 0 Sample Input For Custom Testing STDIN Function ----- 5 → arr[] size n = 5 5 → arr[] = [5, 1, 3, 7, 3] 1 3 7 3 Sample Output 6 Explanation n = 5 arr = [5, 1, 3, 7, 3] If arr is rearranged as arr' = [1, 3, 3, 5, 7], the differences are minimized. The final answer is  $|1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6$ . Sample Case 1 Sample Input For Custom Testing STDIN Function ----- 2 → arr[] size n = 2 3 → arr[] = [3, 2] 2 Sample Output 1 Explanation n = 2 arr = [3, 2] There is no need to rearrange because there are only two elements. The final answer is  $|3 - 2| = 1$ .

**Answer:** (penalty regime: 0 %)

Reset answer

```
1  /*
2   * Complete the 'minDiff' function below.
3   *
4   * The function is expected to return an INTEGER.
5   * The function accepts INTEGER_ARRAY arr as parameter.
6   */
7
8  int minDiff(int arr_count, int* arr)
9  {
10     int sum=0;
11     for(int i=0;i<arr_count;i++){
12         for(int j=i+1;j<arr_count;j++){
13             if(arr[i]>arr[j]){
14                 int temp=arr[i];
15                 arr[i]=arr[j];
16                 arr[j]=temp;
17             }
18         }
19     }
20     for(int i=0;i<arr_count-1;i++){
21         if(arr[i]<arr[i+1])  sum+=(arr[i+1]-arr[i]);
```

```
8 int minDiff(int arr_count, int* arr)
9 {
10     int sum=0;
11     for(int i=0;i<arr_count;i++){
12         for(int j=i+1;j<arr_count;j++){
13             if(arr[i]>arr[j]){
14                 int temp=arr[i];
15                 arr[i]=arr[j];
16                 arr[j]=temp;
17             }
18         }
19     }
20     for(int i=0;i<arr_count-1;i++){
21         if(arr[i]<arr[i+1]) sum+=(arr[i+1]-arr[i]);
22         else sum+=(arr[i]-arr[i+1]);
23     }
24     return sum;
25 }
26 }
```

	Test	Expected	Got	
✓	int arr[] = {5, 1, 3, 7, 3}; printf("%d", minDiff(5, arr))	6	6	✓

Passed all tests! ✓

Finish review

**Duration** 20 mins 22 secsQuestion **1**

Correct

[Flag question](#)

You are transporting some boxes through a tunnel, where each box is a parallelepiped, and is characterized by its length, width and height.

The height of the tunnel **41** feet and the width can be assumed to be infinite. A box can be carried through the tunnel only if its height is strictly less than the tunnel's height. Find the volume of each box that can be successfully transported to the other end of the tunnel. Note: Boxes cannot be rotated.

#### Input Format

The first line contains a single integer ***n***, denoting the number of boxes.

***n*** lines follow with three integers on each separated by single spaces - ***length<sub>i</sub>***, ***width<sub>i</sub>*** and ***height<sub>i</sub>*** which are length, width and height in feet of the ***i*-th** box.

#### Constraints

**$1 \leq n \leq 100$**

**$1 \leq \text{length}_i, \text{width}_i, \text{height}_i \leq 100$**

#### Output Format

For every box from the input which has a height lesser than **41** feet, print its volume in a separate line.

#### Sample Input 0

4

5 5 5

1 2 4 0

10 5 4 1

7 2 4 2

#### Sample Output 0

125

80

#### Explanation 0

The first box is really low, only **5** feet tall, so it can pass through the tunnel and its volume is  **$5 \times 5 \times 5 = 125$** .

The second box is sufficiently low, its volume is  **$1 \times 2 \times 4 = 80$** .

The third box is exactly **41** feet tall, so it cannot pass. The same can be said about the fourth box.

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 struct Box{
```

```

4 int width;
5 int height;
6 };
7 int main()
8 {
9 int n;
10 scanf("%d",&n);
11 struct Box boxes[n];
12 for(int i=0;i<n;i++){
13 scanf("%d %d %d",&boxes[i].length,&boxes[i].width,&boxes[i].height);
14 }
15 for (int i=0;i<n;i++){
16 if(boxes[i].height<41){
17 int volume = boxes[i].length*boxes[i].width*boxes[i].height;
18 printf("%d\n",volume);
19 }
20 }
21 return 0;
22 }

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	4 5 5 5 1 2 40 10 5 41 7 2 42	125 80    	125 80	✓

Passed all tests! ✓