

## Week : 01-01

### Q1) Problem Statement:

This is a simple challenge to help you practice printing to stdout. We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string Hello, World! to stdout.

#### Input Format

You do not need to read any input in this challenge.

#### Output Format

Print Hello, World! to stdout.

#### Sample Output 1

Hello, World!

Status: Finished  
Started: Monday, 22 December 2008, 9:32 PM  
Completed: 5:00 AM, 23 December 2008, 9:18 PM  
Duration: 64 days, 20 hours.

**Question 1**  
Correct  
Attempted out of 100  
Edit question

**Objectives:**  
This is a simple challenge to help you practice printing to stdout.

We're starting out by printing the most famous computing phrase of all time! In the editor below, use either printf or cout to print the string **Hello, World!** to stdout.

**Input Format:**  
You do not need to read any input in this challenge.

**Output Format:**  
Print **Hello, World!** to stdout.

**Sample Output:**  
Hello, World!

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

```
1 #include<cs50.h>
2 int main()
3 {
4     printf("Hello, World!");
5     return 0;
6 }
```

#### OUTPUT:

	Expected	Got	
✓	Hello, World!	Hello, World!	✓

Passed all tests! ✓

Q2) 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);
```

### Sample Input 1

C

program

Programming using C

### Sample Output 1

C

program

Programming using C

#### 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 }
```

### OUTPUT:

	Input	Expected	Got	
✓	C	C	C	✓

Passed all tests! ✓

### Q3) Problem Statement:

The fundamental data types in c are int, float and char. Today, we're discussing int and float data types.

The printf() function prints the given statement to the console. The syntax is printf("format string", argument\_list);. In the function, if we are using an integer, character, string or float as argument, then in the format string we have to write %d (integer), %c (character), %s (string), %f (float) respectively.

The scanf() function reads the input data from the console. The syntax is scanf("format string", argument\_list);. For ex: The scanf("%d", &number) statement reads integer number from the console and stores the given value in variable number. To input two integers separated by a space on a single line, the command is scanf("%d %d", &n, &m), where n and m are the two integers.

### Task

Your task is to take two numbers of int data type, two numbers of float data type as input and output their sum:

Declare 4 variables: two of type int and two of type float.

Read 2 lines of input from stdin (according to the sequence given in the 'Input Format' section below) and initialize your 4 variables.

Use the + and - operator to perform the following operations:

Print the sum and difference of two int variable on a new line.

Print the sum and difference of two float variable 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:  $1 \leq$  integer variables  $\leq 10^4$

$1 \leq$  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

#### Description

The fundamental data types in c are int, float and char. Today, we'll be discussing int and float type.

The printf function prints the given statement to the console. The syntax is printf("format string", argument\_1, argument\_2, ...); in the function. If we are using %d (integer), %f (float) or %c (char) as arguments, then in the format string we have to write %d (integer), %f (float) or %c (char) respectively.

The scanf function reads the user data from the console. The syntax is scanf("format string", argument\_1, argument\_2, ...); the input that function takes must match the given format exactly. It reads the integer value from the console and stores the given value in variable number.

The input size integers is implicitly 4 bytes or a single line. The command is scanf("%d %d", &a, &b); where a and b are variable names.

#### Task

You have to take two inputs of int data type, sum them up and print the sum and difference of both numbers.

1. Declares variables `a` of type int and `b` of type float.
2. Read 2 floats of input from user (according to the requirements given in the Input Format section below) and initialize your variables.
3. Declare a variable `c` and perform the following operations:
  - 4. Compute sum and difference of two int variables stored in `a` and `b`.
  - 5. Compute sum and difference of two float variables rounded to one decimal places and store it in `c`.

#### Input Format

The first line contains two integers.

The second line contains two floating-point numbers.

#### Constraints

$1 \leq a \leq 10^9$   
 $1 \leq b \leq 10^9$

#### Output Format

Print the sum and difference of both integers separated by a space in the first line and the sum and difference of both float variables separated by a space in the second line.

#### Sample Input

```
10 20  
3.14 1.59
```

#### Sample Output

```
30 10  
4.71 2.68
```

#### Explanation

Since we sum the integers 10 and 20 we get the integer 30. When we subtract the second integer from the first we get 8.0 from 10-2.0=8.0.

Since we sum the floating-point numbers 3.14 and 1.59 we get 4.71 when we sum the second number with the first. And we get 2.68 as the difference.

```
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,a-b);  
9     printf("\n%.1f %.1f",c+d,c-d);  
10    return 0;  
11 }
```

## OUTPUT:

	Input	Expected	Got	
✓	10 4 6.6 2.0	14 6 6.6 2.0	14 6 6.6 2.0	✓
✗	10 0 3.3 1.0	10 12 12.0 4.0	10 12 12.0 4.0	✗

Passed all tests! ✓

## Week:01-02

### Q1) Problem Statement

Write a program to input a name (as a single character) and marks of three tests as m1, m2, and m3 of a student considering all the three marks have been given in integer format.

Now, you need to calculate the average of the given marks and print it along with the name as mentioned in the output format section.

All the test marks are in integers and hence calculate the average in integer as well. That is, you need to print the integer part of the average only and neglect the decimal part.

**Input Format:**

Line 1: Name (Single character)

Line 2: Marks scored in the 3 tests separated by single space.

**Output Format:**

First line of output prints the name of the student. Second line of the output prints the average mark.

**Constraints**

Marks for each student lie in the range 0 to 100 (both inclusive)

**Sample Input 1:**

A

3 4 6

**Sample Output 1:**

A

4



```

1 #include <stdio.h>
2 int main()
3 {
4     char ch;
5     int m1,m2,m3;
6     scanf("%c %d %d", &ch, &m1, &m2);
7     m3=(m1+m2)/2;
8     printf("%c %d %d", ch, m1, m2);
9     return 0;
10 }

```

## OUTPUT:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	3 4 6	3 4	3 4	✓
✓	T 7 8	T 6	T 6	✓
✓	R 3 100 99 66	R 3 100 99 66	R 3 100 99 66	✓

Passed all testcases ✓

## Q2) Problem Statement:

Some C data types, their format specifiers, and their most common bit widths are as follows:

Int ("%d"): 32 Bit integer

Long ("%ld"): 64 bit integer

Char ("%c"): Character type

Float ("%f"): 32 bit real value

Double ("%lf"): 64 bit real value

### Reading

To read a data type, use the following syntax:

scanf("format\_specifier", &val)

For example, to read a character followed by a double:

C

char ch;

double d;

scanf("%c %lf", &ch, &d);

For the moment, we can ignore the spacing between format specifiers.

### Printing

To print a data type, use the following syntax:

```
printf("formatSpecifier", val)
```

For example, to print a character followed by a double:

C

```
char ch = 'd';  
double d = 234.432;  
printf("%c %lf", ch, d);
```

Note: You can also use cin and cout instead of scanf and printf; however, if you are taking a million numbers as input and printing a million lines, it is faster to use scanf and printf.

### Input Format

Input consists of the following space-separated values: int, long, char, float, and double, respectively.

### Output Format

Print each element on a new line in the same order it was received as input. Note that the floating-point value should be correct up to 3 decimal places and the double to 9 decimal places.

### Sample Input

```
3  
12345678912345  
a  
334.23  
14049.30493
```

### Sample Output

```
3  
12345678912345  
a  
334.230  
14049.304930000
```

**Ques 2**

Given 2 float values, their decimal components, and their integer components are as follows:

- float("3.14", 3.14f) → 3.14f

**Reading:**  
The code is written like this. The following code:  

```
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
```

 It has statements that are printing the existing maximum floating-point numbers.

**Writing:**  
The code is written like this. The following code:  

```
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
float("3.14", 3.14f);
```

 It has statements that are printing the existing maximum floating-point numbers.

**Output:**  
The output is given below:  

```
3.1400000000000002
3.1400000000000002
3.1400000000000002
3.1400000000000002
3.1400000000000002
3.1400000000000002
```

**Sample Output:**  
3.1400000000000002

**OUTPUT:**

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	3 12345678912345 a 334.23 14049.30493	3 12345678912345 a 334.230 14049.304930000	3 12345678912345 a 334.230 14049.304930000	✓

Passed all tests! ✓

### Q3) Problem Statement:

Write a program to print the ASCII value and the two adjacent characters of the given character.

**Input Format:** Reads the character

**Output Format:** First line prints the ascii value, second line prints the previous character and next character of the input character

**Sample Input 1:**

E

**Sample Output 1:**

69

DF

**Question 3**

Correct

Marked out of  
7.00[Flag question](#)

Write a program to print the ASCII value and the two adjacent characters of the given character.

Input

E

Output

69

D F

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

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

**OUTPUT:**

	Input	Expected	Got	
✓	E	69 D F	69 D F	✓

Passed all tests! ✓

# Week 2

## Week-02-01-Practice Session-Coding

**Q1)** Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.

Hint:

One foot is 12 inches.

One inch is 2.54 centimeters.

Input Format

First line, read the number of feet.

Second line, read the number of inches.

Output Format

In one line print the height in centimeters.

Note: All of the values should be displayed using two decimal places.

Sample Input 1

5 6

Sample Output 1

167.64

```
1 #include <stdio.h>
2
3 int main() {
4     int feet, inch;
5     scanf("%d", &feet);
6     scanf("%d", &inch);
7
8     float cm = ((12*feet) + inch) * 2.54;
9     printf("%.2f", cm);
10 }
```

<b>Status:</b>	Finished
<b>Started:</b>	Monday, 23 December 2024, 5:33 PM
<b>Completed:</b>	Wednesday, 18 October 2024, 3:11 PM
<b>Duration:</b>	88 days, 2 hours
<b>Question 1</b>	
Scored:	
Marks out of:	100
If Preparation:	
Many people think about their height in feet and inches, even in some countries that primarily use the metric system. Write a program that reads a number of feet from the user, followed by a number of inches. Once these values are read, your program should compute and display the equivalent number of centimeters.	
Hint:	
One foot is 12 inches.	
One inch is 2.54 centimeters.	
Input Format:	
First line: read the number of feet.	
Second line: read the number of inches.	
Output Format:	
To one line print the height in centimeters.	
Note: All of the values should be displayed using two decimal places.	
Sample Input 1	
5 6	
Sample Output 1	
167.64	

## Output:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	5	167.64	167.64	✓
	6			

Passed all tests! ✓

**Q2)** Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

#### Input Format

First line, read the first number.

Second line, read the second number.

#### Output Format

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a

Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b

Fifth line, print the remainder when a is divided by b

#### Sample

Input 1 100 6

#### Sample Output

106 94 600 16 4



Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- The sum of a and b
- The difference when b is subtracted from a
- The product of a and b
- The quotient when a is divided by b
- The remainder when a is divided by b

#### Input Format

First line, read the first number.

Second line, read the second number.

#### Output Format

First line, print the sum of a and b

Second line, print the difference when b is subtracted from a

Third line, print the product of a and b

Fourth line, print the quotient when a is divided by b

Fifth line, print the remainder when a is divided by b

#### Sample

Input 1 100 6

#### Sample Output

106 94 600 16 4

```
1 #include <stdio.h>
2
3 int main() {
4     int n1, n2;
5     scanf("%d", &n1);
6     scanf("%d", &n2);
7
8     printf("%d\n%d\n%d\n%d\n%d", (n1+n2), (n1-n2), (n1*n2), (n1/n2), (n1%n2));
9 }
```

Ouput:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	100	106	106	✓
	6	94	94	
	600	600	600	
	16	16	16	
	4	4	4	

Passed all tests! ✓

**Q3)** A bakery sells loaves of bread for \$3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

#### Input Format

Read the number of day old loaves.

#### Output Format

First line, print Regular price: price

Second line, print Discount: discount

Third line, print Total: total

Note: All of the values should be displayed using two decimal places.

#### Sample Input 1

10

#### Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

Question 3  
Correct  
Marked out of  
100  
Type question

A bakery sells loaves of bread for \$3.49 each. Day old bread is discounted by 60 percent. Write a program that begins by reading the number of loaves of day old bread being purchased from the user. Then your program should display the regular price for the bread, the discount because it is a day old, and the total price. Each of these amounts should be displayed on its own line with an appropriate label. All of the values should be displayed using two decimal places.

Input Format

Read the number of day old loaves.

Output Format

First line, print Regular price: price

Second line, print Discount: discount

Third line, print Total: total

Note: All of the values should be displayed using two decimal places.

Sample Input 1

10

Sample Output 1

Regular price: 34.90

Discount: 20.94

Total: 13.96

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     float p = 3.49;
8     float disP = (p*n) * (60.0/100.0);
9     float tP = p * n;
10    float fP = tP - disP;
11
12    printf("Regular price: %.2f\nDiscount: %.2f\nTotal: %.2f", tP, disP, fP);
13
14
15 }
```

#### Output:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	10	Regular price: 34.90 Discount: 20.94 Total: 13.96	Regular price: 34.90 Discount: 20.94 Total: 13.96	✓

Passed all tests! ✓

# **Week-02-02-Practice Session-Coding**

Q1) 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

Status: Finished  
Started: Monday, 23 December 2024, 5:33 PM  
Completed: Wednesday, 16 October 2024, 3:23 PM  
Duration: 51 days, 2 hours.

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 ( $Y < X \leq N$ ). 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

$1 \leq N \leq 100000$

$1 \leq X, Y \leq 100000$

SAMPLE INPUT 1

100 110

SAMPLE OUTPUT 1

YES

SAMPLE INPUT 2

100 90

SAMPLE OUTPUT 2

NO

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

## Output:

	Input	Expected	Got	
✓	100 110	YES	YES	✓
✓	100 90	NO	NO	✓

Passed all tests! ✓

**Q2)** 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

$0 < N < 10^6$

#### 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.

Question 2  
Correct  
Marked out of  
5/5  
P. Pugnacious

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

$0 < N < 10^6$

#### 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.

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int s = n*(n - 1)/2;
8     printf("%d", s);
9 }
```

**Output:**

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1	0	0	✓
✓	2	1	1	✓

Passed all tests! ✓

**Q3)** 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.



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 team 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.

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

**Output:**

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	81 26 15	81	81	✓

Passed all tests! ✓

# Coding-Variables and Keywords

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Wednesday, 16 October 2024, 3:48 PM
Duration	68 days 1 hour

Q1) Read the code given below to learn naming conventions in identifiers.

For example, consider the program given below:

```
#include <stdio.h>
```

```
int main()
{
    int age = 2; // age is an integer variable

    int firstNumber = 2; // firstNumber is an integer variable
```

// If there are two or more words in an identifier/variable - User can also use "camel case" style to declare a variable.

```
int second_number = 3; // second_number is an integer variable
```

// Any space cannot be used between two words of an identifier/variable; User can use underscore (\_) instead of space.

```
int _i_am_also_a_valid_identifier = 4; // _i_am_also_a_valid_identifier is an integer variable
```

// An identifier/variable name must be start with an alphabet or underscore (\_) only, no other special characters, digits are allowed as first character of the identifier/variable name.

```
printf("age = %d\n", age);
printf("firstNumber = %d\n", firstNumber);
printf("second_number = %d\n", second_number);
printf("_i_am_also_a_valid_identifier = %d\n", _i_am_also_a_valid_identifier);
return 0;
}
```

Fill in the missing code in the below program to print the values of the given variables.

Question 1  
Correct  
Marked out of  
1.00  
Type your answer

Read the code given below to learn naming conventions in identifiers.  
For example, consider the program given below:  

```
#include <stdio.h>

int main()
{
    int age = 2; // age is an integer variable

    int firstNumber = 3; // firstNumber is an integer variable

    // If there are two or more words in an Identifier/variableName then we have to "separate" them by underscores(_). Instead of space.

    int second_number = 3; // second_number is an integer variable

    // An underscore cannot be used between the words of an Identifier/variableName; User can use underscore(_) instead of space.

    int _i_am_also_a_valid_identifier = 4; // _i_am_also_a_valid_identifier is an integer variable

    // An Identifier/variableName must start with an alphabet or underscore (_), not with other special characters; digits are allowed as first character of the Identifier/variable name.

    printf("age = %d\n", age);
    printf("firstNumber = %d\n", firstNumber);
    printf("second_number = %d\n", second_number);
    printf("_i_am_also_a_valid_identifier = %d\n", _i_am_also_a_valid_identifier);
    return 0;
}
```

Fill in the missing code in the below program to print the values of the given variables.

Answer: (penalty regime 0%)

Reset answer

```
#include <stdio.h>

int main()
{
    int age = 2;
    int firstNumber = 3;
    int second_number = 3;
    int _i_am_also_a_valid_identifier = 4;
    printf("age = %d\n", age); // FILL in the missing code
    printf("firstNumber = %d\n", firstNumber); // FILL in the missing code
    printf("second_number = %d\n", second_number); // FILL in the missing code
    printf("_i_am_also_a_valid_identifier = %d\n", _i_am_also_a_valid_identifier); // FILL in the missing code
    return 0;
}
```

Output:

	Expected	Got	
✓	age = 2 firstNumber = 2 second_number = 3 _i_am_also_a_valid_identifier = 4	age = 2 firstNumber = 2 second_number = 3 _i_am_also_a_valid_identifier = 4	✓

Passed all tests! ✓

# Coding-Syntax of main() function

Q1) Identify and correct the error in the code given below.

Status: Finished  
Started: Monday, 23 December 2024, 5:33 PM  
Completed: Wednesday, 16 October 2024, 1:50 PM  
Duration: 68 days 1 hour

Question 1  
Correct:  
Marked out of 1.00  
1.00  
Flag question  
Reset answer

Identify and correct the error in the code given below.

Answer: (penalty regime: 0 %)

```
1 #include <csfdev.h>
2
3 int main()
4 {
5     printf("Hello, # is a preprocessor in C");
6     return 0;
7 }
```

Output:

	Expected	Got	
✓	Hello, # is a preprocessor in C	Hello, # is a preprocessor in C	✓

Passed all tests! ✓

**Q2)** In **C** programming language, execution of the code starts with a **function** called **main**.

We shall learn more about functions in the later sections. For now, we can safely assume that **function** is the name given to a set of one or more executable statements. **main()** is a **user defined function**, i.e., a user (a programmer) writes the code for the **main()** function.

While executing a **C** program, the **Operating System (OS)** only calls the **main()** function in that program.

When the **OS** executes a program, the program usually returns an integer value 0 if the execution of that program is successful.

In **C**, **main()** can be written in such a way that it returns a **an int**.

```
#include <stdio.h>
```

```
int main()
{
    printf("Sample main() function with int as return type!");
    return 0;// 0 value indicates that the execution is successful

}
```

If the programmer does not specify any return type, the return type is by default considered as **int**.

The name of the **main()** function should always be in lowercase, i.e., if a function is written as **Main()**, it is not the **main** function which is called by the **OS**.

Read the code given below to familiarize yourself with the syntax of **main()** function. Retype in the space provided.

```
#include <stdio.h>

int main()
{
    printf("Impossible is nothing!");
    return 0;
}
```

Question 2  
Correct  
Solved out of 1.00  
Flag question

In C programming language, execution of the code starts with a function called `main`.

We shall learn more about functions in the later sections. For now, we can safely assume that `function` is the name given to a set of one or more executable statements. `main()` is a **user defined function**, i.e., a user (a programmer) writes the code for the `main()` function.

While executing a C program, the **Operating System (OS)** only calls the `main()` function in that program.

When the OS executes a program, the program usually returns an integer value 0 if the execution of that program is successful.

In C, `main()` can be written in such a way that it returns an int.

```
#include <stdio.h>

int main()
{
    printf("Hello world() function with int as return type!");
    return 0 // 0 value indicates that the execution is successful.
}
```

If the programmer does not specify any return type, the return type is by default considered as int.

The name of the `main()` function should always be in lowercase, i.e., if a function is written as `Main()`, it is not the `main` function which is called by the OS.

Read the code given below to familiarize yourself with the syntax of `main()` function. Retype in the space provided.

```
#include <stdio.h>

int main()
{
    printf("Impossible is nothing!");
    return 0;
}
```

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

```
1 #include <stdio.h>
2
3 v int main() {
4     printf("Impossible is nothing!");
5 }
```

**Output:**

	Expected	Got	
✓	Impossible is nothing!	Impossible is nothing!	✓

Passed all tests! ✓

Q3) Identify and correct the error in the code given below.

Question 3

Correct

Marked out of  
1.00

Flag question

Identify and correct the error in the code given below.

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

[Reset answer](#)

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, I am learning C Language!");
6     return 0;
7 }
```

Expected	Got	
✓ Hello, I am learning C Language!	Hello, I am learning C Language!	✓

Passed all tests! ✓

Q4) Click on Check without correcting the code.

This results in many errors because the main function is not defined correctly.

Now, correct the spelling of the main function and submit the program once again.

Question 4

Correct

Marked out of  
1.00

Flag question

Click on **Check** without correcting the code.

This results in many errors because the main function is not defined correctly.

Now, correct the spelling of the main function and submit the program once again.

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

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Correct Me!");
6     return 0;
7 }
```

	Expected	Got	
✓	Correct Me!	Correct Me!	✓

Passed all tests! ✓

Q5) Identify and correct the error in the code given below.

Question 5

Correct

Marked out of  
1.00

Flag question

Identify and correct the error in the code given below.

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

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, float data type allocates 4 bytes in memory");
6     return 0;
7 }
```

Expected	Got	
✓ Hello, float data type allocates 4 bytes in memory	Hello, float data type allocates 4 bytes in memory	✓

Passed all tests! ✓

# int Data Type

Q1) In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.

Read the code given below and retype in the space provided.

```
#include <stdio.h>
int main()
{
    int binaryThree = 0b11;
    printf("binaryThree value = %d\n", binaryThree);
    int octalEight = 010;
    printf("octalEight value = %d\n", octalEight);
    int hexTen = 0xA;
    printf("hexTen value = %d\n", hexTen);
    int asciiValueOfOne = '1';
    printf("asciiValueOfOne value = %d\n", asciiValueOfOne);
    int asciiValueOfA = 'A';
    printf("asciiValueOfA value = %d\n", asciiValueOfA);
    return 0;
}
```

Status: Finished  
Started: Monday, 23 December 2024, 5:03 PM  
Completed: Wednesday, 16 October 2024, 3:54 PM  
Duration: 88 days, 1 hour

Question #  
Done:  
Marked out of  
1.00  
1 Requested

In the program given below, we shall learn how to assign values to int data type from binary, octal, hex and character literals.

Read the code given below and retype in the space provided.

```
#include <stdio.h>
int main()
{
    int binaryThree = 0b11;
    printf("binaryThree value = %d\n", binaryThree);
    int octalEight = 010;
    printf("octalEight value = %d\n", octalEight);
    int hexTen = 0xA;
    printf("hexTen value = %d\n", hexTen);
    int asciiValueOfOne = '1';
    printf("asciiValueOfOne value = %d\n", asciiValueOfOne);
    int asciiValueOfA = 'A';
    printf("asciiValueOfA value = %d\n", asciiValueOfA);
    return 0;
}
```

Answer: [penalty regime 0 %]

```
1 #include <stdio.h>
2
3 int main() {
4     int b1 = 0b11;
5     printf("binaryThree value = %d\n", b1);
6     int o8 = 010;
7     int h10 = 0xA;
8     printf("octalEight value = %d\n", o8);
9     printf("hexTen value = %d\n", h10);
10    int a1 = '1';
11    int aA = 'A';
12    printf("asciiValueOfOne value = %d\n", a1);
13    printf("asciiValueOfA value = %d\n", aA);
14 }
```

**Output:**

	<b>Expected</b>	<b>Got</b>	
✓	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	binaryThree value = 3 octalEight value = 8 hexTen value = 10 asciiValueOfOne value = 49 asciiValueOfA value = 65	✓

Passed all tests! ✓

Q2) In the program given below, fill in the missing code to add two integer numbers.

Question 2

Correct

Marked out of  
1.00

Flag question

In the program given below, fill in the missing code to add two integer numbers.

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

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int num1 = 15, num2 = 25, sum;
6     printf("Given integers are num1 = %d, num2 = %d\n", num1, num2);
7     sum = num1+num2;
8     printf("Sum of 2 given numbers = %d\n", sum);
9     return 0;
10 }
```

Expected	Got
✓ Given integers are num1 = 15, num2 = 25 Sum of 2 given numbers = 40	Given integers are num1 = 15, num2 = 25 ✓ Sum of 2 given numbers = 40

Passed all tests! ✓

**Q3)** To print unsigned values on the console, use %u format character instead of %d in the **printf()** function.

Whenever an attempt is made to assign a negative number to an **unsigned int** ( For eg: `unsigned int num = -1;`) the compiler does not flag it as an **error**. Instead, it will automatically convert the negative number to a positive number as shown below:

```
unsigned int num = -1;
```

The value stored in **num** = `unsigned int maximum_value + 1 - num;`

The final value in **num** = 4294967295 (in a 32-bit processing system)

In the program given below, fill in the missing format characters to print signed and unsigned values.

Question 3  
Direct  
Marked out of  
1.00  
P Flag question

To print unsigned values on the console, use %u format character instead of %d in the **printf()** function.

Whenever an attempt is made to assign a negative number to an **unsigned int** ( For eg: `unsigned int num = -1;`) the compiler does not flag it as an **error**. Instead, it will automatically convert the negative number to a positive number as shown below:

```
unsigned int num = -1;  
The value stored in num = unsigned int maximum_value + 1 - num;  
The final value in num = 4294967295 (in a 32-bit processing system)
```

In the program given below, fill in the missing **format characters** to print signed and unsigned values.

Answer: (empty string) 0%

Reset answer

```
#include <iostream>  
int main()  
{  
    signed int number1 = -10; number2 = 20;  
    unsigned int number3 = -1; number4 = 21;  
    printf("Given signed values are %d and %d\n", number1, number2); // Fill the correct format character after %  
    printf("Given unsigned values are %u and %u\n", number3, number4); // Fill the correct format character after %  
    return 0;  
}
```

Expected	Get
<input checked="" type="checkbox"/> Given signed values are -10 and 20 <input checked="" type="checkbox"/> Given unsigned values are 4294967295 and 1	<input checked="" type="checkbox"/> Given signed values are -10 and 20 <input checked="" type="checkbox"/> Given unsigned values are 4294967295 and 1

Passed all tests: ✓

**Q4) Identify the error and correct the code. [Hint: Verify if all variables are declared before they are first used.]**

Question 4

Correct

Marks out of  
1.00

1 Mark question

Identify the error and correct the code. [Hint: Verify if all variables are declared before they are first used.]

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

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     Int number1 = 20, number2 = 20;
6     Int sub = number1 - number2;
7     printf("The difference of the two given numbers = %d\n", sub);
8     return 0;
9 }
```

	Expected	Got	
<input checked="" type="checkbox"/>	The difference of the two given numbers = -10	The difference of the two given numbers = -10 <input checked="" type="checkbox"/>	

Passed all test! ✓

# float Data Type

Q1) Identify and correct the errors in the code given below:

Expected Output:

Given float values are num1 = 5.340000, num2 = 125.789001

The result after dividing in float format = 23.555992

The result after dividing in exponential format = 2.355599e+01

Status: Finished  
Started: Monday, 23 December 2024, 5:35 PM  
Completed: Wednesday, 16 October 2024, 3:56 PM  
Duration: 66 days 1 hour

Question 1  
Coded  
Marked out of 1.00  
Flag question

Identify and correct the errors in the code given below.  
Expected Output:  
Given float values are num1 = 5.340000, num2 = 125.789001  
The result after dividing in float format = 23.555992  
The result after dividing in exponential format = 2.355599e+01

Answer: (corrected) 0.50

Reset answer

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float num1 = 5.34, num2 = 125.789, result;
6     printf("Given float values are num1 = %f, num2 = %f\n", num1, num2);
7     result = num2 / num1;
8     printf("The result after dividing in float format = %f\n", result);
9     printf("The result after dividing in exponential format = %e\n", result);
10    return 0;
11 }
```

Expected	Got
<input checked="" type="checkbox"/> Given float values are num1 = 5.340000, num2 = 125.789001 The result after dividing in float format = 23.555992 The result after dividing in exponential format = 2.355599e+01	<input checked="" type="checkbox"/> Given float values are num1 = 5.340000, num2 = 125.789001 The result after dividing in float format = 23.555992 The result after dividing in exponential format = 2.355599e+01

Passed all tests! ✓

**Q2) Identify and correct the errors in the code given below:**

Submit 2  
Correct  
Marked out of 100  
By my teacher

Identify and correct the errors in the code given below.

**Answer: (penalty regime D78)**

Reset answer

```
1 float calc_D78(float num1, float num2)
2 {
3     float result;
4     if(num1 == 0.0 || num2 == 0.0)
5     {
6         cout << "Error! Both values are zero." << endl;
7         exit(1);
8     }
9     result = num1 / num2;
10    cout << "Result of division = " << result;
11    return 0;
12 }
```

Expected	Got
Both the values are num1 = 5.3456, num2 = 11.499999 Result of division = 0.462946	Both float values are num1 = 5.34566, num2 = 11.499999 result of division = 0.462946 ✓

Reset all tests ✓

# Cricket Stadium

Q1) There was a large ground in center of the city which is rectangular in shape. The Corporation decides to build a Cricket stadium in the area for school and college students, But the area was used as a car parking zone. In order to protect the land from using as an unauthorized parking zone, the corporation wanted to protect the stadium by building a fence. In order to help the workers to build a fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help. Can you please help them by writing a program to find the exact length of the rope and the exact quantity of carpet that is required?

**Input format:**

Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.

**Output Format:**

Output Consists of two integers. The first integer corresponds to the length. The second integer corresponds to the quantity of carpet required.

**Sample Input:**

50

20

**Sample Output:**

140

1000

For example:

Input	Result
50	140
20	1000

**Status:** Finished  
**Started:** Monday, 25 December 2014, 5:53 PM  
**Completed:** Wednesday, 16 October 2014, 4:17 PM  
**Duration:** 61 days 1 hour

Question 1  
Solved  
Marked out of 100  
0%  
For example:

There was a large ground in center of the city which is rectangular in shape. The Corporation decides to build a Cricket stadium in the area for school and college students. But the area was used as a car parking zone. In order to protect the land from using as an unauthorized parking zone, the corporation wanted to protect the stadium by building a fence. In order to help the workers to build a fence, they planned to place a thick rope around the ground. They wanted to buy only the exact length of the rope that is needed. They also wanted to cover the entire ground with a carpet during rainy season. They wanted to buy only the exact quantity of carpet that is needed. They requested your help. Can you please help them by writing a program to find the exact length of the rope and the exact quantity of carpet that is required?

Input Format:

Input consists of 2 integers. The first integer corresponds to the length of the ground and the second integer corresponds to the breadth of the ground.

Output Format:

Output Consists of two integers. The first integer corresponds to the length. The second integer corresponds to the quantity of carpet required.

Sample Input:

50

39

Sample Output:

140

1000

For example:

Input	Result
50	140
39	1000

Answer: (penalty regime 0%)

```
1 package com.intellisys;
2
3 public class Main {
4     public static void main(String[] args) {
5         int width = 50;
6         int height = 39;
7         System.out.println("Length: " + width);
8         System.out.println("Breadth: " + height);
9         System.out.println("Perimeter: " + (width * 2) + (height * 2));
10    }
11 }
```

## Output:

	Input	Expected	Got	
✓	50	140	140	✓
	20	1000	1000	

Passed all tests! ✓

## Sports Day Celebration

Q1) Training for sports day has begun and the physical education teacher has decided to conduct some team games. The teacher wants to split the students in higher secondary into equal sized teams. In some cases, there may be some students who are left out from the teams and he wanted to use the left out students to assist him in conducting the team games. For instance, if there are 50 students in a class and if the class has to be divided into 7 equal sized teams, 7 students will be there in each team and 1 student will be left out. That 1 student will assist the PET. With this idea in mind, the PET wants your help to automate this team splitting task. Can you please help him out?

**INPUT FORMAT:**

Input consists of 2 integers. The first integer corresponds to the number of students in the class and the second integer corresponds to the number of teams.

**OUTPUT FORMAT:**

The output consists of two integers. The first integer corresponds to the number of students in each team and the second integer corresponds to the students who are left out.

**SAMPLE INPUT:**

60

8

**SAMPLE OUTPUT:**

7

4

For example:

Input	Result
60	7
8	4

**Status**: Finished  
**Started**: Monday, 23 December 2024, 5:35 PM  
**Completed**: Wednesday, 16 October 2024, 4:19 PM  
**Duration**: 6 days 8 hours

**Question 1**  
 Score: 1  
 Marks out of 1.00  
[Flag question](#)

Training for sports day has begun and the physical education teacher has decided to conduct some team games. The teacher wants to split the students in Higher secondary into equal sized teams. In some cases, there may be some students who are left out from the teams and he wanted to use the left out students to assist him in conducting the team games. For instance, if there are 50 students in a class and if the class has to be divided into 7 equal sized teams, 7 students will be there in each team and 1 student will be left out. That 1 student will assist the P.E. With this idea in mind, the P.E wants your help to automate this team splitting task. Can you please help him out?

**INPUT FORMAT:**  
 Input consists of 2 integers. The first integer corresponds to the number of students in the class and the second integer corresponds to the number of teams.

**OUTPUT FORMAT:**  
 The output consists of two integers. The first integer corresponds to the number of students in each team and the second integer corresponds to the students who are left out.

**SAMPLE INPUT:**  
 60  
 8

**SAMPLE OUTPUT:**  
 7  
 4

**For example:**

Input	Result
40	7
8	4

**Answer:** (penalty regime: 0.50)

```

1 #include <stdio.h>
2
3 int main() {
4     int p, t;
5     scanf("%d", &p);
6     scanf("%d", &t);
7     printf("Marked %d (%d), (%d)\n",
8         p/t, (p%t));
9 }
```

Output:

	Input	Expected	Got	
✓	60	7	7	✓
	8	4	4	

Passed all tests! ✓

# The Newspaper Agency

Q1) Each Sunday, a newspaper agency sells  $w$  copies of a special edition newspaper for Rs. $x$  per copy. The cost to the agency of each newspaper is Rs. $y$ . The agency pays a fixed cost for storage, delivery and so on of Rs.100 per Sunday. The newspaper agency wants to calculate the profit which it obtains only on Sundays. Can you please help them out by writing a program to compute the profit if  $w$ ,  $x$ , and  $y$  are given?

**INPUT FORMAT:**

Input consists of 3 integers:  $w$ ,  $x$ , and  $y$ .  $w$  is the number of copies sold,  $x$  is the cost per copy and  $y$  is the cost the agency spends per copy.

**OUTPUT FORMAT:**

The output consists of a single integer which corresponds to the profit obtained by the newspaper agency.

**SAMPLE INPUT:**

1000

2

1

**SAMPLE OUTPUT:**

900

For example:

Input	Result
1000	
2	900
1	

**Status:** Finished  
**Started:** Monday, 23 December 2024, 5:33 PM  
**Completed:** Wednesday, 16 October 2024, 4:21 PM  
**Duration:** 68 days, 1 hour

Question 1  
Career  
Marked out of 1.00  
Y. Pop Culture

Each Sunday, a newspaper agency sells  $w$  copies of a special edition newspaper for  $R$  rupees per copy. The cost to the agency of each newspaper is  $R_1$ . The agency pays a fixed cost for storage, delivery and so on of Rs.100 per Sunday. The newspaper agency wants to calculate the profit which it obtains only on Sundays. Can you please help them out by writing a program to compute the profit if  $w$ ,  $x$  and  $y$  are given?

INPUT FORMAT:

Input consists of 3 integers:  $w$ ,  $x$ , and  $y$ .  $w$  is the number of copies sold,  $x$  is the cost per copy and  $y$  is the cost the agency spends per copy.

OUTPUT FORMAT:

The output consists of a single integer which corresponds to the profit obtained by the newspaper agency.

SAMPLE INPUT:

1000

2

1

SAMPLE OUTPUT:

900

For example:

Input	Result
1000	900
2	
1	

Answer: (penalty regime: 2%)

```
1. #include <iostream>
2.
3. int main() {
4.     int w, x, y;
5.     scanf("%d", &w);
6.     scanf("%d", &x);
7.     scanf("%d", &y);
8.
9.     int k = (w * x) - (y + 100);
10.    printf("%d\n", k);
11. }
```

## Output:

	Input	Expected	Got	
✓	1000 2 1	900	900	✓

Passed all tests! ✓

## The Chronicles of Narnia

Q1) Four kids Peter, Susan, Edmond and Lucy travel through a wardrobe to the land of Narnia. Narnia is a fantasy world of magic with mythical beasts and talking animals. While exploring the land of Narnia Lucy found Mr. Tumnus the two-legged stag, and she followed it, down a narrow path. She and Mr. Tumnus became friends and he offered a cup of coffee to Lucy in his small hut. It was time for Lucy to return to her family and so she bid good bye to Mr. Tumnus and while leaving Mr. Tumnus told that it is quite difficult to find the route back as it was already dark. He told her to see the trees while returning back and said that the first tree with two digits number will help her find the way and the way to go back to her home is the sum of digits of the tree and that numbered way will lead her to the tree next to the wardrobe where she can find the others. Lucy was already confused, so please help her in finding the route to her home....

**Input Format:**

Input consists of an integer corresponding to the 2-digit number.

**Output Format:**

Output consists of an integer corresponding to the sum of its digits.

**SAMPLE INPUT:**

87

**SAMPLE OUTPUT:**

15

For example:

Input	Result
87	15

**Status:** Finished  
**Started:** Monday, 23 December 2014, 5:33 PM  
**Completed:** Wednesday, 16 October 2014, 4:22 PM  
**Duration:** 08 days 1 hour

Question 1  
Completed  
Marked out of 100  
100% correct

Four kids Peter, Susan, Edmund, and Lucy travel through a wardrobe to the land of Narnia. Narnia is a fantasy world of magic with mythical beasts and talking animals. While exploring the land of Narnia Lucy found Mr. Tumnus the two-legged stag, and she followed it down a narrow path. She and Mr. Tumnus became friends and he offered a cup of coffee to Lucy in his small hut. It was time for Lucy to return to her family and so she bid good bye to Mr. Tumnus and while leaving Mr. Tumnus told that it is quite difficult to find the route back as it was already dark. He told her to see the trees while returning back and said that the first tree with two digits number will help her find the way and the way to go back to her home is the sum of digits of the tree and that number's way will lead her to the tree next to the wall which she can find the others. Lucy was already confused, so please help her in finding the route to her home...

Input Format:

Input consists of an integer corresponding to the 2-digit number.

Output Format:

Output consists of an integer corresponding to the sum of its digits.

SAMPLE INPUT:

87

SAMPLE OUTPUT:

15

For example:

Input	Result
87	15

Answer: (Initially regime 0 %)

```
1 //Write code here
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int p = n;
8
9     while (n != 0) {
10         p += (n % 10);
11         n /= 10;
12     }
13
14     printf("%d", p);
15 }
```

## Output:

	Input	Expected	Got	
✓	87	15	15	✓
✓	54	9	9	✓

Passed all tests! ✓

## Week:03-01

### Q1) Problem Statement:

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

Status: Finished  
Started: Monday, 23 December 2014, 5:33 PM  
Completed: Monday, 4 November 2014, 2:18 PM  
Duration: 49 days 5 hours

Question 1  
Concept:  
Marked out of 3.00  
T. Play question

Answer: (Openly mine 0.50)

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int num1,num2;
6     scanf("%d %d",&num1,&num2);
7     int lastdigit1 = num1%10;
8     int lastdigit2 = num2%10;
9     if(lastdigit1 == lastdigit2)
10    {
11        printf("true");
12    }
13    else
14    {
15        printf("false");
16    }
17    return 0;
18 }
```

	Input	Expected	Got
1	25 53	false	false ✓
2	27 77	true	true ✓

Passed all tests! ✓

### Q2) Problem Statement:

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

$1 < n < 100$

## Output Format

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

### Sample Input 0

3

### Sample Output 0

Weird

The screenshot shows a programming challenge interface with the following details:

- Description:** *Print Weird* - Write a program which prints "Weird" if the given integer n is weird, otherwise print "Not Weird".
- Test:** Given an integer n, print the following based on whether the number is weird:
  - If it's odd, print **Weird**.
  - If it's even and in the inclusive range of 2 to 5, print **Not Weird**.
  - If it's even and in the inclusive range of 6 to 20, print **Weird**.
  - If it's even and greater than 20, print **Not Weird**.
- Completion:** Complete the provided code template to print the correct output for any given input.
- Input Format:** A single line containing a positive integer n.
- Output Format:** Print **Weird** if the number is weird; otherwise, print **Not Weird**.
- Sample Input 0:** 3
- Sample Output 0:** Weird
- Execution:**
  - Sample Output 0: Weird
  - Actual Output: Weird
- Message:** All test cases passed.

```

1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n%2==0)
7     {
8         printf("Weird");
9     }
10    else
11    {
12        if(n>2 && n<5)
13        {
14            printf("Not Weird");
15        }
16        else if(n>5 && n<20)
17        {
18            printf("Not Weird");
19        }
20        else if(n>20)
21        {
22            printf("Not Weird");
23        }
24    }
25 }
```

OUTPUT:

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

Passed all tests! ✓

### Q3) Problem Statement:

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  $3^2 + 4^2 = 5^2$

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

Question 3

Correct

Marked out of  
1.00

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 6 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) || (a*a+c*c==b*b)) || ((b*b+c*c==a*a))
7         printf("yes");
8     else
9         printf("no");
10    return 0;
11 }
```

OUTPUT:

	Input	Expected	Got	
✓	3 5 4	yes	yes	✓
✓	5 6 2	no	no	✓

Passed all tests! ✓

## Week:03-02

### Q1) Problem Statement:

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.

Status: Pending  
Started: Monday, 21 November 2023, 2:03 PM  
Completed: Monday, 6 November 2023, 6:08 PM  
Duration: an hour (40 minutes)

Submit | Correct | Mark as not done | Delete | Report

Write a program that determines the name of a shape based on 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.

```

1 #include <stdio.h>
2 
3 int main()
4 {
5     int sides;
6     scanf("%d", &sides);
7     if(sides < 3)
8         printf("triangle\n");
9     else if(sides == 4)
10        printf("square\n");
11    else if(sides == 5)
12        printf("pentagon\n");
13    else if(sides == 6)
14        printf("hexagon\n");
15    else if(sides == 7)
16        printf("heptagon\n");
17    else if(sides == 8)
18        printf("octagon\n");
19    else if(sides == 9)
20        printf("nonagon\n");
21    else if(sides == 10)
22        printf("decagon\n");
23    else if(sides == 11)
24        printf("undecagon\n");
25    else if(sides == 12)
26        printf("dodecagon\n");
27    else
28        printf("The number of sides is not supported.\n");
29    return 0;
30 }

```

OUTPUT:

	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! ✓

## Q2) Problem Statement:

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

```
The Chinese-zodiac animals for years in a 12-year cycle. Only 100-year cycle is shown in the table below. The zodiac repeats from Hare with 2012 being another year of the Dragon, and 1999 being another year of the Tiger.

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 programme that reads a year from the user and displays the animal associated with that year. Your programme should work correctly for any year greater than or equal to zero, and print that value stored in the variable 'year'.
```

Sample Input 1  
2004  
Sample Output 1  
Monkey  
Sample Input 2  
2010  
Sample Output 2  
Tiger

```
3 #include<stdio.h>
4 int main()
5 {
6     int year;
7     const char*animals[] = {
8         "Dragon", "Snake", "Horse", "Sheep", "Monkey", "Rooster",
9         "Dog", "Pig", "Rat", "Ox", "Tiger", "Hare"
10    };
11    scanf("%d", &year);
12    int index = (year - 2000) % 12;
13    if(index < 0)
14    {
15        index += 12;
16    }
17    printf("%s\n", animals[index]);
18 }
```

OUTPUT:

	Input	Expected	Got	
✓	2004	Monkey	Monkey	✓
✓	2010	Tiger	Tiger	✓

Passed all tests! ✓

### Q3) Problem Statement:

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

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 2

D 5

Sample Output 2

The square is white.

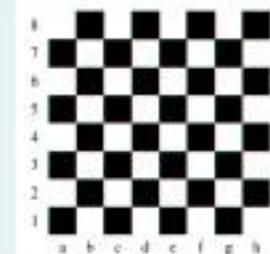
Question 3

Cleared

Marked out of  
100

1/1 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 2

d 5

Sample Output 2

The square is white.

```
1 #include<stdio.h>
2 int main()
3 {
4     char column;
5     int row;
6     scanf("%c%d",&column,&row);
7     int columnindex = column-'a';
8     if((columnindex+row)%2==0)
9     {
10         printf("The square is white.\n");
11     }
12     else
13     {
14         printf("The square is black.\n");
15     }
16     return 0;
17 }
```

OUTPUT:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	a 1	The square is black.	The square is black.	✓
✓	d 5	The square is white.	The square is white.	✓

Passed all tests! ✓

## Week:03-03

### Q1) Problem Statement:

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

2020

#### Sample Output 1

170

Status: Finished  
Started: Monday, 23 December 2024, 5:33 PM  
Completed: Thursday, 28 November 2024, 9:55 PM  
Duration: 24 days, 19 hours

Question 1  
Done  
Marked out of 100  
P. Hey person

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  
2020

Sample Output 1

170

```

1 int daysInYear(int year)
2 {
3     if (year % 400 == 0)
4         return 366;
5     else if (year % 100 == 0)
6         return 365;
7     else if (year % 4 == 0)
8         return 366;
9     else
10        return 365;
11 }
12
13 int daysInMonth(int day, int month, int year)
14 {
15     const int daysInMonth[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 31, 30};
16     if (day > 31)
17         return -1;
18     else
19         return daysInMonth[month - 1];
20 }
21
22 void solve()
23 {
24     cout << daysInMonth(15, 1, 2020);
25 }
26
27 int main()
28 {
29     solve();
30     return 0;
31 }

```

## OUTPUT:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	18 6 2020	170	170 ✓	

Passed all tests! ✓

## Q2) Problem Statement:

Suppandi is trying to take pan in the local village math qua. 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 I 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 -> Rectangle, S--> Square, T->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

Suspank is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suspank is confused; he was never very 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.

Assume he is confused; he just says something random, at this point, all you can do is say to help Suspank to print the correct answer in an integer.

Input Format:

- Name of shape (shape is in upper case R is rectangle, S is square, T is 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

```
1
T
10
20
```

Sample Output 1

```
200
```

```
1 #include<stdio.h>
2 int calculatearea(char shape,int length1,int length2)
3 {
4     int area=0;
5     switch(shape)
6     {
7         case'R':
8             area=length1*length2;
9             break;
10        case'S':
11            area=(length1*length2)/2;
12            break;
13        case'T':
14            area=length1*length2;
15            break;
16        default:
17            area=0;
18            break;
19    }
20    return area;
21 }
22 int main()
23 {
24     char shape;
25     int length1,length2;
26     scanf("%c",&shape);
27     scanf("%d%d",&length1,&length2);
28     int area=calculatearea(shape,length1,length2);
29     printf("%d\n",area);
30 }
```

## OUTPUT:

	Input	Expected	Got	
✓	T 10 20	200	200	✓
✓	S 30 40	600	600	✓
✓	B 2 11	0	0	✓
✓	R 10 30	300	300	✓
✓	S 40 50	1000	1000	✓

Passed all tests! ✓

## Q3) Problem Statement:

superday he arrives there. They to his home planet. It is very important for him to know which day of the week with the four. They don't follow a 10-day following days: hitney follow

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

Sample Input

7

Sample Output

Kryptonday

Sample Input

1

Sample Output

Monday

Submit 3  
Output  
Memory usage: 7.16  
Time  
Top submissions

Kryptonite 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 day's Day Number:

Name of Day: 1 Sunday, 2 Monday, 3 Tuesday, 4 Wednesday, 5 Thursday, 6 Friday, 7 Saturday, 8 Kryptonday, 9 Colossay, 10 Disasterday. Here are the rules of the 10-day week - The calendar starts with Sunday always. • It has only 200 days, after the 200th 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 <= 200)

Output Format: Print the name of the day you are arriving on.

Example Input  
7

Example Output  
Kryptonday

Example Input  
1

Example Output  
Sunday

Answer: Identify regime (0 to 9)  
1. 

```
public class Main {
    public static void main(String[] args) {
        int n = 7;
        System.out.println(day(n));
    }
    static String day(int n) {
        String[] days = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Kryptonday", "Colossay", "Disasterday"};
        return days[n];
    }
}
```

OUTPUT:

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

Passed all tests! ✓



## **WEEK 4**

Q1) 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     int T,i=0,n,t;
4     scanf("%d",&T);
5     while(i<T){
6         scanf("%d",&n);
7         t=n/4;
8         if(t%2==0 && n%2==0){
9             printf("No\n");
10        }else if(t%2==1 && n%2==1){
11            printf("No\n");
12        }else{
13            printf("Yes\n");
14        }
15        i++;
16    }
17 }
```

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

Passed all tests! ✓

Q2) 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

630

#### 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  $0 + 0 + 2 + 2 = 4$ .

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

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

	Input	Expected	Got	
✓	638	2	2	✓
✓	1288	4	4	✓

Passed all tests! ✓

Q3) 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<=T<=100 1<=N<=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. According to Manish {\$1, \$2, \$3,... \$10} must be distributed. But as per Manisha only {\$1, \$2, \$3, \$4} coins are enough to purchase any item ranging from \$1 to \$10. Hence minimum is 4.

Likewise denominations could also be {\$1, \$2, \$3, \$5}. Hence answer is still 4.

For test case 2, N=5. According to Manish {\$1, \$2, \$3, \$4, \$5} must be distributed. But as per Manisha only {\$1, \$2, \$3} coins are enough to purchase any item ranging from \$1 to \$5. Hence minimum is 3. Likewise, denominations could also be {\$1, \$2, \$4}. Hence answer is still 3.

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

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

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓
✓	1000	10	10	✓

Passed all tests! ✓

**Q4)** 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     int n,x=0;
4     while(scanf("%d",&n)==1){
5         if(n%2!=0){
6             x++;
7         }
8         printf("%d",x);
9     }
10    return 0;
11 }
```

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

Passed all tests! ✓

Q5) 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  $86 \neq 89$ .

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

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

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

Passed all tests! ✓

Q6) 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.

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 (109 + 7).

It has the following:

n: an integer that denotes the number of food items

k: an integer that denotes the unhealthy number

Constraints

$$1 \leq n \leq 2 \times 10^9 \quad 1 \leq k \leq 4 \times 10^15$$

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 43 3

Explanation - The following sequence of n = 2 food items: 1. 2. Item 1 has 1 macronutrients.  $1 + 2 = 3$ ; observe that this is the max total, and having avoided having exactly k = 2 macronutrients.

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

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

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

Passed all tests! ✓

## WEEK 5

Q) Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.

The lines after that contain different values for size of the chessboard

Output format:

Print a chessboard of dimensions size \* size. Print a W for white spaces and B for black spaces.

Input:

2

3

5

Output:

WBW

BWB

WBW

WBWBW

BWBWB

WBWBW

BWBWB

WBWBW

```

1 #include <iostream.h>
2 int main(){
3     int t,s;
4     scanf("%d",&t);
5     for (int a=0;a<t;a++){
6         scanf("%c",&s);
7         for(int i=0;i<s;i++){
8             for (int j=0;j<5;j++){
9                 if((i+j)%2==0){
10                     printf("Q");
11                 }else{
12                     printf("R");
13                 }
14             }
15             printf("\n");
16         }
17     }
18     return 0;
19 }
```

	Input	Expected	Got	
✓	2	RR QQ	RR QQ	✓
	1	RR	RR	
	5	RRR RRRRR RRRRR RRRRR RRRRR RRRRR RRRRR RRRRR	RRR RRRRR RRRRR RRRRR RRRRR RRRRR RRRRR RRRRR	

Passed all tests! ✓

Q) Let's print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases

Each test case contains an integer N and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

2 W

3 B

Output:

WB

BW

BWB

WBW

BWB

```
1 #include <stdio.h>
2 int main(){
3     int t,d,i,j,k,s,z;
4     char c,s;
5     scanf("%d", &t);
6     for(i=0;i<t;i++){
7         for(j=0;j<i;j++){
8             scanf("%d %c", &d,&s);
9             for(k=j;k<i;k++){
10                 z=(s-'W')*10+1;
11                 d=(j*2+1)*10+1;
12                 for(l=k;l<d;l++){
13                     z=(l*2-s)*10+1;
14                     printf("%c", c);
15                 }
16             }
17         }
18     }
19     return 0;
20 }
```

	Input	Expected	Got	
1	2	WB	WB	✓
2	3	BW	BW	
3	4	BWB	BWB	
		WBW	WBW	
		BWB	BWB	

Passed all tests! ✓

Q) Decode the logic and print the Pattern that corresponds to given input.

If N= 3

then pattern will be :

10203010011012

\*\*4050809

\*\*\*\*607

If N= 4, then pattern will be:

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*10011

Constraints

$2 \leq N \leq 100$

Input Format

First line contains T, the number of test cases

Each test case contains a single integer N

Output

First line print Case #i where i is the test case number

In the subsequent line, print the pattern

Test Case 1

3

3

4

5

Output

Case #1

10203010011012

\*\*4050809

\*\*\*\*607

Case #2

1020304017018019020

\*\*50607014015016

\*\*\*\*809012013

\*\*\*\*\*10011

Case #3

102030405026027028029030

\*\*6070809022023024025

\*\*\*\*10011012019020021

\*\*\*\*\*13014017018

\*\*\*\*\*15016

```

1 #include <iostream.h>
2 int main(){
3     int n,v,p,c,in,i,j,k,t,ti;
4     scanf("%d",&t);
5     for(ti=0;ti<t;ti++){
6         v=0;
7         scanf("%d",&n);
8         printf("Case #%-4d:\n",ti+1);
9         for(i=0;i<n;i--){
10             c=0;
11             if(i>0){
12                 for(j=0;j<i;j++){
13                     printf("%c",v);
14                 }
15             }
16             for(j=i;j<n;j++){
17                 if(j>0) c++;
18                 printf("%c",v+c);
19             }
20             if(i>0){
21                 p=v+(v*(v-1))/2;
22                 in=p;
23             }
24             in=in-c;
25             p=in;
26             for(k=1;k<p;k--){
27                 printf("%c",p++);
28                 if(k==n-1) printf("\n");
29             }
30         } printf("\n");
31     }
32 }
33 }
```

	Input	Expected	Got
✓	1. Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	✓
✗	2. Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	✗
✗	3. Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	Case #1 10200010012002 ***102000001 *****10200 Case #2 10200000100100010000 ***10000010010000 *****100001100100 *****100011 Case #3 1020000001001000100000 ***100000001001000000 *****100011001000000000 *****100100010000000000 *****15816	✗

Passed all tests! ✓

Q) The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an Armstrong number.

Example 1:

Input:

153

Output:

true

Explanation:

153 is a 3-digit number, and  $153 = 1^3 + 5^3 + 3^3$ .

Example 2:

Input:

123

Output:

false

Explanation:

123 is a 3-digit number, and  $123 \neq 1^3 + 2^3 + 3^3 = 36$ .

Example 3:

Input:

1634

Output:

true

Note:

$1 \leq N \leq 10^8$

```
1 #include <stdio.h>
2 #include <math.h>
3 int main(){
4     int n;
5     scanf("%d", &n);
6     int x=0, n2=n;
7     while(n2>0){
8         x++;
9         n2=n2/10;
10    }
11    int sum=0;
12    int n4=n,x4;
13    while(x4>0){
14        n4=n4*10;
15        sum=sum+pow(n4,x);
16        n3=n3/10;
17    }
18    if(n==sum){
19        printf("true");
20    } else{
21        printf("false");
22    }
23    return 0;
24 }
```

	Input	Expected	Got	
✓	1634	true	true	✓
✓	123	false	false	✓

Passed all tests! ✓

Q) Take a number, reverse it and add it to the original number until the obtained number is a palindrome. Constraints 1<=num<=99999999 Sample Input 1 32 Sample Output1 55

Sample Input 2 789 Sample Output 2 66066

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

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

Passed all tests! ✓

**Q)** 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

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

	Input	Expected	Got	
✓	34	33344	33344	✓

Passed all tests! ✓

## WEEK 6

### Q1) Objective

In this challenge, we're going to use loops to help us do some simple math. Check out the Tutorial tab to learn more.

#### Task

Given an integer,  $n$ , print its first **10** multiples. Each multiple  $n \times i$  (where  $1 \leq i \leq 10$ ) should be printed on a new line in the form:  $n \times i = \text{result}$ .

#### Input Format

A single integer,  $n$ .

#### Constraints

**$2 \leq n \leq 20$**

#### Output Format

Print **10** lines of output; each line  $i$  (where  $1 \leq i \leq 10$ ) contains the **result** of  $n \times i$  in the form:

$n \times i = \text{result}$ .

#### Sample Input

2

#### Sample Output

$2 \times 1 = 2$

$2 \times 2 = 4$

$2 \times 3 = 6$

$2 \times 4 = 8$

$2 \times 5 = 10$

$2 \times 6 = 12$

$2 \times 7 = 14$

$2 \times 8 = 16$

$2 \times 9 = 18$

$2 \times 10 = 20$

```

1 //Simple calculator
2 int main() {
3     int n, result, i=1;
4     scanf("%d", &n);
5     while (i<=n) {
6         result=i*i;
7         printf("%d * %d = %d\n", i, i, result);
8         printf("\n");
9         i++;
10    }
11 }

```

	Input	Expected	Get	
✓	1	$2 \times 1 = 2$	$2 \times 1 = 2$	✓
		$2 \times 2 = 4$	$2 \times 2 = 4$	
		$2 \times 3 = 6$	$2 \times 3 = 6$	
		$2 \times 4 = 8$	$2 \times 4 = 8$	
		$2 \times 5 = 10$	$2 \times 5 = 10$	
		$2 \times 6 = 12$	$2 \times 6 = 12$	
		$2 \times 7 = 14$	$2 \times 7 = 14$	
		$2 \times 8 = 16$	$2 \times 8 = 16$	
		$2 \times 9 = 18$	$2 \times 9 = 18$	
		$2 \times 10 = 20$	$2 \times 10 = 20$	

Passed all testcases!

Q2) A nutritionist is labelling 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.

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

- $1 \leq n \leq 2 \times 10^9$
- $1 \leq k \leq 4 \times 10^{15}$

### 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

#### Sample Output 2

5

#### Explanation 2

$2 + 3 = 5$ , is the best case for maximum nutrients.

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

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

Passed all tests! ✓

Q3) Determine all positive integer values that evenly divide into a number, its factors. Return the  $p^{th}$  element of your list, sorted ascending. If there is no  $p^{th}$  element, return 0.

For example, given the number  $n = 20$ , its factors are  $\{1, 2, 4, 5, 10, 20\}$ . Using **1-based indexing** if  $p = 3$ , return 4. If  $p > 6$ , return 0.

Complete the code in the editor below. The function should return a long integer value of the  $p^{th}$  integer factor of  $n$ .

It has the following:

$n$ : an integer

$p$ : an integer

### 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 Input 0

10

3

### Sample Output 0

5

### Explanation 0

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . We then return the  $p = 3^{rd}$  factor as our answer.

### Sample Input 1

10

5

### Sample Output 1

0

### Explanation 1

Factoring  $n = 10$  we get  $\{1, 2, 5, 10\}$ . There are only 4 factors and  $p = 5$ . We return 0 as our answer.

### Sample Input 2

1

1

### Sample Output 2

1

### Explanation 2

Factoring  $n = 1$  we get  $\{1\}$ . We then return the  $p = 1^{st}$  factor as our answer.

```
1 #include <cs50.h>
2 int main() {
3     int a[100], n, count=0, p=1;
4     string s;
5     scanf("%s", &s);
6     for (int i=1; i<n; i++) {
7         if (n/i==0) {
8             a[count]=i;
9             count++;
10        }
11    }
12    for (int i=0; i<(count-1); i++) {
13        for (int j=(i+1); j<count; j++) {
14            if (a[i]==a[j]) {
15                t=a[i];
16                a[i]=a[j];
17                a[j]=t;
18            }
19        }
20    }
21    if (p>0 || p==count) {
22        printf("%d", a[p-1]);
23    } else {
24        printf("0");
25    }
26    return 0;
27 }
```

	Input	Expected	Got
✓	10 1	5	5 ✓
✓	10 5	0	0 ✓
✓	1 1	1	1 ✓

Passed all tests! ✓

## WEEK 7

Q) 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
- $\text{cost}$ : an integer array denoting the cost of each flavor of ice cream

### Input Format

The first line contains an integer,  $t$ , denoting the number of trips to the ice cream parlor. The next  $t$  sets of lines each describe a visit. Each trip is described as follows:

1. The integer  $m$ , the amount of money they have pooled.

2. The integer  $n$ , the number of flavors offered at the time.
3.  $n$  space-separated integers denoting the cost of each flavor:  $\text{cost}[\text{cost}[1], \text{cost}[2], \dots, \text{cost}[n]]$ .

**Note:** The index within the cost array represents the flavor of the ice cream purchased.

## Constraints

- $1 \leq t \leq 50$
- $2 \leq m \leq 10^4$
- $2 \leq n \leq 10^4$
- $1 \leq \text{cost}[i] \leq 10^4$ , " $i \in [1, n]$ "
- There will always be a unique solution.

## Output Format

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  $\mathbf{1 + 3 = 4}$ .
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  $\mathbf{2 + 2 = 4}$ .

```

1 #include <cs50.h>
2 int main(){
3     int t,n,i,C=0;
4     scanf("%d", &t);
5     for(int i=0;i<t;i++){
6         C=0;
7         scanf("%d%d", &n, &m);
8         int arr[n];
9         for(int j=0;j<n;j++){
10             scanf("%d", &arr[j]);
11         }
12         for(int a=0;a<n-1;a++){
13             for(int b=a+1;b<n;b++){
14                 if(arr[a]>arr[b]==m){
15                     printf("%d %d\n", a+1, b+1);
16                     C+=1; break;
17                 }
18             }
19         }
20     }
21 }
22

```

	Input	Expected	Got	
✓	2	1 4	1 4	✓
	4	1 2	1 1	
	1			
	1 6 5 3 2			
	a..			
	a..			
	2 2 4 3			

Passed all tests! ✓

Q) Numeros the Artist had two lists that were permutations of one another. He was very proud. Unfortunately, while transporting them from one exhibition to another, some numbers were lost out of the first list. Can you find the missing numbers?

As an example, the array with some numbers missing, **arr = [7, 2, 5, 3, 5, 3]**. The original array of numbers **brr = [7, 2, 5, 4, 6, 3, 5, 3]**. The numbers missing are **[4, 6]**.

## Notes

- If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.

- You have to print all the missing numbers in ascending order.
- Print each missing number once, even if it is missing multiple times.
- The difference between maximum and minimum number in the second list is less than or equal to **100**.

Complete the code in the editor below. It should return an array of missing numbers.

It has the following:

- arr: the array with missing numbers
- brr: the original array of numbers

## Input Format

There will be four lines of input:

**n** - the size of the first list, **arr**

The next line contains **n** space-separated integers **arr[i]**

**m** - the size of the second list, **brr**

The next line contains **m** space-separated integers **brr[i]**

## Constraints

- **$1 \leq n, m \leq 2 \times 10^5$**
- **$n \leq m$**
- **$1 \leq brr[i] \leq 2 \times 10^4$**
- **$X_{max} - X_{min} < 101$**

## Output Format

Output the missing numbers in ascending order.

## Sample Input

10

203 204 205 206 207 208 203 204 205 206

13

203 204 204 205 206 206 207 205 208 203 206 205 206 204

## Sample Output

204 205 206

## Explanation

**204** is present in both arrays. Its frequency in ***arr*** is **2**, while its frequency in ***brr*** is **3**. Similarly, **205** and **206** occur twice in ***arr***, but three times in ***brr***. The rest of the numbers have the same frequencies in both lists.

```

1 #include <iostream>
2 int main(){
3     int n,m,c,cl=0,cr;
4     scanf("%d",&n);
5     int ar[n];
6     for(int a=0;a<n;a++){
7         scanf("%d",&ar[a]);
8     }
9     scanf("%d",&m);
10    int br[m],ar[m];
11    for(int b=0;b<m;b++){
12        scanf("%d",&br[b]);
13    }
14    for(int j=0;j<m;j++){
15        c=0;
16        for(int i=0;i<n;i++){
17            if(ar[i]==br[j]){
18                c+=1;
19                ar[i]=-1;
20                break;
21            }
22        }
23        if(c==0){
24            ar[c1]=br[j];
25            c1+=1;
26        }
27    }
28    for(int a=0;a<c1;a++){
29        co=0;
30        for(int b=0;b<cl;b++){
31            if(ar[b]==ar[a]){
32                co++;
33            }
34        }
35        int temp=ar[a];
36        ar[a]=ar[co];
37        ar[co]=temp;
38    }
39    for(int i=0;i<c1;i++){
40        printf("%d ",ar[i]);
41    }
42    return 0;
}

```

	Input	Expected	Got	
✓	10 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205 206 204	204 205 206 204 205 206	204 205 206	✓

Passed all tests! ✓

Q) Watson gives Sherlock an array of integers. His challenge is to find an element of the array such that the sum of all elements to the left is equal to the sum of all elements to the right. For instance, given the array **arr = [5, 6, 8, 11]**, **8** is between two subarrays that sum to **11**. If your starting array is **[1]**, that element satisfies the rule as left and right sum to **0**.

You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Complete the code in the editor below. It should return a string, either YES if there is an element meeting the criterion or NO otherwise.

It has the following:

- arr: an array of integers

### Input Format

The first line contains  $T$ , the number of test cases.

The next  $T$  pairs of lines each represent a test case.

- The first line contains  $n$ , the number of elements in the array  $arr$ .
- The second line contains  $n$  space-separated integers  $arr[i]$  where  $0 \leq i < n$ .

### Constraints

- $1 \leq T \leq 10$
- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq 2 \times 10^4$
- $0 \leq i \leq n$

### Output Format

For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

### Sample Input 0

2

3

1 2 3

4

1 2 3 3

### **Sample Output 0**

NO

YES

### **Explanation 0**

For the first test case, no such index exists.

For the second test case,  $\text{arr}[0] + \text{arr}[1] = \text{arr}[3]$ , therefore index **2** satisfies the given conditions.

### **Sample Input 1**

3

5

1 1 4 1 1

4

2 0 0 0

4

0 0 2 0

### **Sample Output 1**

YES

YES

YES

## Explanation 1

In the first test case, **arr[2] = 4** is between two subarrays summing to **2**.

In the second case, **arr[0] = 2** is between two subarrays summing to **0**.

In the third case, **arr[2] = 2** is between two subarrays summing to **0**.

```
1 #include <stdio.h>
2 int main(){
3     int t,n,ls,rs,m;
4     scanf("%d",&t);
5     for(int i=0;i<t;i++){
6         ls=0;
7         rs=0;
8         scanf("%d",&n);
9         int ar[n];
10        for(int j=0;j<n;j++){
11            scanf("%d",&ar[j]);
12        }
13        m=n/2;
14        if(ar[m]==0){
15            for(m=0;ar[m]==0&&m<n;m++);
16        }
17        for(int j=0;j<m;j++)
18            ls+=ar[j];
19        for(int j=m;j<n;j++)
20            rs+=ar[j];
21        printf("%c\n", (ls==rs)? "YES" : "NO");
22    }
23    return 0;
24 }
```

	Input	Expected	Got	
✓	3 5 1 1 4 1 1 4 2 0 0 0 4 0 0 2 0	YES YES YES YES	YES YES YES	✓
✓	2 3 1 2 3 4 1 2 3 3	NO YES	NO YES	✓

Passed all tests! ✓

# WEEK 08 – CODING

## Question 1 :

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:

$1 \leq t \leq 10$   $1 \leq n \leq 1000$   $1 \leq a[i] \leq 1000$

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  $(2+4+5=)14$ . Minimum sum using the 4 elements

would be  $(1+2+3+4=)10$ . Difference will be  $14-10=4$

```

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 }
34 }
35 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1 5 1 1 2 3 4 5	4	4	✓

Passed all tests! ✓

## **Question 2**

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

```

1 #include<stdio.h>
2 int main()
3 {
4     int n,min1,min2,temp,flag=-1;
5     scanf("%d",&n);
6     int vac[n],pat[n];
7     for(int i=0;i<n;i++)
8         scanf("%d",&vac[i]);
9     for(int i=0;i<n;i++)
10        scanf("%d",&pat[i]);
11
12     for(int j=0;j<n-1;j++)
13     {
14         min1=j,min2=j;
15         for(int k=j;k<n;k++)
16         {
17             if(vac[k]<vac[min1])
18                 min1=k;
19             if(pat[k]<pat[min2])
20                 min2=k;
21         }
22         temp=vac[min1];
23         vac[min1]=vac[j];
24         vac[j]=temp;
25
26         temp=pat[min2];
27         pat[min2]=pat[j];
28         pat[j]=temp;
29
30     }
31     for(int i=0;i<n;i++)
32     {
33         if(vac[i]<=pat[i])
34         {
35             flag=0;
36             break;
37         }
38     }
39     if(flag==1)
40         printf("Yes");
41     else
42         printf("No");
43 }
44 }
```

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

Passed all tests! ✓

### **Question 3**

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  $(2, 5)$

```
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++)
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 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	5 1 3 1 4 3	2	2	✓

Passed all tests! ✓

## **Question 4**

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:**

$1 \leq m \leq 106$

$0 \leq A[i] \leq 106$

**NOTE:** The indexing of the array starts with 0.

**SAMPLE INPUT**

5

4 5 3 7 1

**SAMPLE OUTPUT**

4 2 0 1 3

```

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 |    }
16 |    max++;
17 |    int min=0;
18 |    for(int a=0;a<n;a++)
19 |    {
20 |        for(int b=0;b<n;b++)
21 |        {
22 |            if(arr[b]<arr[min])
23 |                min=b;
24 |
25 |        }
26 |        printf("%d ",min);
27 |        arr[min]=max;
28 |    }
29 |

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	5 4 5 3 7 1	4 2 0 1 3	4 2 0 1 3	✓

## Question 1

### Question text

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

### 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

**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;
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 }
25 }
```

	Input	Expected	Got
✓	1 2 3 4 5 6 7 8 9	25 28	25 28
✓	21 422 423 443 586 645 657 846 904	2591 2356	2591 2356

Passed all tests! ✓

## Question 2

### Question text

Shyam Lal, a wealthy landlord from the state of Rajasthan, being an old fellow and tired of doing hard work, decided to

Sell all his farmland and to live rest of his life with that money. No other farmer is rich enough to buy all his land so heDecided to partition the land into rectangular plots of different sizes with different cost per unit area. So, he sold thesePlots to the farmers but made a mistake. Being illiterate, he made partitions that could be overlapping. When the Farmers came to know about it, they ran to him for compensation of extra money they paid to him. So, he decided to Return all the money to the farmers of that land which was overlapping with other farmer's land to settle down theConflict. All the portion of conflicted land will be taken back by the landlord.To decide the total compensation, he has to calculate the total amount of money to return back to farmers with the Same cost they had purchased from him. Suppose, Shyam Lal has a total land area of 1000 x 1000 equal square blocks Where each block is equivalent to a unit square area which can be represented on the co-ordinate axis. Now find the Total amount of money, he has to return to the farmers. Help Shyam Lal to accomplish this task.

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int i,j,n,x1,x2,y1,y2,t=0;
6     long long total=0;
7     int arr[1001][1001]={0};
8     scanf("%d",&n);
9     while(n--)
10    {
11        scanf("%d %d %d %d %d",&x1,&y1,&x2,&y2,&t);
12        for(i=x1;i<x2;i++)
13        {
14            for(j=y1;j<y2;j++)
15            {
16                if(arr[i][j]==0)
17                    arr[i][j]=t;
18                else if(arr[i][j]>0)
19                    arr[i][j]-=(-1)*(arr[i][j]+t);
20                else if(arr[i][j]<0)
21                    arr[i][j]=-t;
22            }
23        }
24    }
25    for(i=1;i<1001;i++)
26    {
27        for(j=1;j<1001;j++)
28        {
29            if(arr[i][j]<0)
30                total+=arr[i][j];
31        }
32    }
33
34    printf("%lld\n",(-1)*total);
35    return 0;
36 }
```

	Input	Expected	Got	
✓	3 3 4 4 9 3 4 3 6 6 2 2 2 5 4 3	35	35	✓
✓	1 48 12 49 27 8	0	0	✓
✓	3 88 34 99 76 44 82 65 94 100 81 58 16 65 66 7	10500	10500	✓

Passed all tests! ✓

### Question 3

#### Question text

Microsoft has come to hire interns from your college. N students got shortlisted out of which few were males and a few Females. All the students have been assigned talent levels. Smaller the talent level, lesser is your chance to be selected. Microsoft wants to create the result list where it wants the candidates sorted according to their talent levels, but therels a catch. This time Microsoft wants to hire female candidates first and then male candidates.The task is to create a list where first all-female candidates are sorted in a descending order and then male candidates Are sorted in a descending order.

#### Input Format

The first line contains an integer N denoting the number of students. Next, N lines contain two space-separated Integers, ai and bi.The first integer, ai will be either 1(for a male candidate) or 0(for female candidate)The second integer, bi will be the candidate's talent level.

The image shows a code editor with a C program and a test case table.

```
1 #include<stdio.h>
2
3 struct data
4 {
5     int gen,int tal;
6 };
7
8 int main()
9 {
10    int n;
11    scanf("%d",&n);
12    struct data a[n];
13    for(int i=0;i<n;i++)
14        scanf("%d %d",&a[i].gen,&a[i].tal);
15    for(int i=0;i<n-1;i++)
16    {
17        for(int j=0;j<n-i-1;j++)
18        {
19            if(a[j].tal<a[j+1].tal)
20            {
21                struct data temp=a[j];
22                a[j]=a[j+1];
23                a[j+1]=temp;
24            }
25        }
26    }
27    for(int i=0;i<n;i++)
28    {
29        if(a[i].gen==0)
30            printf("%d ",a[i].tal);
31        }for(int i=0;i<n;i++)
32    {
33        if(a[i].gen==1)
34            printf("%d ",a[i].tal);
35    }
36 }
```

	Input	Expected	Get
1	5 0 1 1 5 0 4 0 7 1 25	7 3 2 25 5	7 3 1 15 8
2	6 0 1 0 26 0 10 0 37 0 7 0 11	39 37 36 13 7 3	39 37 18 13 7 3
3	12 1 21 1 28 1 18 1 13 1 11 1 15 1 19 1 17 1 16 1 14 1 12	31 29 28 18 12 8 1 1 1 1 2 1	31 29 18 14 12 28 8 5 1 2 1
4	10 0 10 1 10 0 10 1 10 0 12 0 12 1 10 0 10 1 10 0 10 1 10	10 10 10 0 10 10 0 10 0 10 10 0 10 10 0 10 0 10 0 10	10 10 10 0 10 10 0 10 0 10 10 0 10 10 0 10 0 10 0 10

Below the table, a green bar indicates "Passed all tests".



# Week 10

Q1)

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

**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**

a114720516

**Sample Output 0**

0 2 1 0 1 1 1 1 0 0

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char str[1000];
6     int freq[10] = {0}; // Initialize array to store frequencies of digits
7
8     scanf("%s", str);
9
10    for (int i = 0; i < strlen(str); i++) {
11        if (str[i] >='0' && str[i] <='9') {
12            int digit = str[i] - '0';
13            freq[digit]++;
14        }
15    }
16
17    for (int i = 0; i < 10; i++) {
18        printf("%d ", freq[i]);
19    }
20
21    return 0;
22 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	a11472o5t6	0 2 1 0 1 1 1 1 0 0	0 2 1 0 1 1 1 1 0 0	✓
✓	1m4n88j12n1	0 2 1 0 1 0 0 0 2 0	0 2 1 0 1 0 0 0 2 0	✓
✓	1v88886125633Bar0ekk	1 1 1 2 0 1 2 0 5 0	1 1 1 2 0 1 2 0 5 0	✓

Passed all tests! ✓

Q2) 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 Format:**

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 Format:**

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

**Constraints:**

$1 \leq T \leq 10$

$1 \leq \text{length of string} \leq 105$

**Sample Input**

2

nBBZLaosnm

JHklsnZtTL

**Sample Output**

2

1

Difficulty: 2  
Solved:  
Marked out of 100  
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:

$1 \leq T \leq 10^3$

$T \times \text{length of string} \leq 10^6$

#### SAMPLE INPUT

```
2
aBcDacomm
#Hehe2T%
```

#### SAMPLE OUTPUT

```
2
1
```

```
1 #include <iostream.h>
2 #include <iostream.h>
3
4 int main() {
5     int T;
6     scanf("%d", &T);
7
8     while (T--) {
9         char str[100005];
10        scanf("%s", str);
11
12        set count = 0;
13        for (int i = 0; i < strlen(str); i++) {
14            char ch = str[i];
15            if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U' ||
16                ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
17                count++;
18            }
19        }
20
21        printf("%d\n", count);
22    }
23
24    return 0;
25 }
```

Output:

	Input	Expected	Got	
✓	2 nBBZLao\$ne JHkIsnZtTL	2 1	2 1	✓
✓	2 nBBZLao\$ne JHkIsnZtTL	2 1	2 1	✓

Passed all tests! ✓

Q3) 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

This is C

#### Sample Output

This

is

C

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

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char sentence[1000];
6
7     // Read the sentence
8     fgets(sentence, sizeof(sentence), stdin);
9
10    // Iterate over the sentence and print each word
11    char *word = strtok(sentence, " ");
12    while (word != NULL) {
13        printf("%s\n", word);
14        word = strtok(NULL, " ");
15    }
16
17    return 0;
18 }
```

Output:

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	This is C	This is C	This is C	✓
✓	Learning C is fun	Learning C is fun	Learning C is fun	✓

Passed all tests! ✓

#### Q4) 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

abcd

ef

#### Sample Output

4 2

abcdef

ebcd af

**Question 4**  
Incorrect  
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

```
abcd
ef
```

#### Sample Output

```
4 2
abcdef
ebcd ef
```

Answers (Accuracy regime 0%)

```
#include <iostream>
using namespace std;
int main()
{
    string str1,str2;
    cout << "Enter string 1" << endl;
    cin >> str1;
    cout << "Enter string 2" << endl;
    cin >> str2;
    cout << str1.length() << " " << str2.length();
    cout << endl;
    cout << str1 << str2;
    cout << endl;
    cout << str2[0] << str1[0];
    cout << endl;
    cout << str1[0] << str2[0];
}
```

Output:

	Input	Expected	Get	
abcd ef	4 2 abcdef ebcd ef	4 2 abcdef ebcd ef	4 2 abcdef ebcd ef	

Passed all tests! ✓

# Week 11

**Q1)** 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

$1 \leq \text{len}(A) \leq 1000000$

$1 \leq \text{len}(B) \leq 1000000$

## 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.

**Status:** Finished  
**Started:** Monday, 23 December 2013, 11:18 PM  
**Completed:** Monday, 23 November 2014, 11:09 PM  
**Duration:** 20 days 5 hours

Question 8  
Solved  
Updated over 6  
days  
130  
17% complete

Two strings A and B consisting 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 `agpt` and we select the prefix `agp` then we can convert it to `ay` by increasing the alphabetical value by 1. But if we select the prefix `ay` then we cannot increase the alphabetical value.

You have 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:

$1 \leq \text{len}(A) \leq 1000000$   
 $1 \leq \text{len}(B) \leq 1000000$

#### SAMPLE INPUT

abccaa  
ccbaa

#### SAMPLE OUTPUT

NO

#### Explanation

The string above can be converted to `abccba` one more and to `ccbab` in the next move.

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdbool.h>
4
5 bool are_compatible(char a[], char b[]) {
6     int lenA = strlen(a);
7     int lenB = strlen(b);
8
9     if (lenA != lenB) {
10         return false;
11     }
12
13     for (int i = 0; i < lenA; i++) {
14         if (a[i] > b[i]) {
15             return false; // characters in A cannot be decreased in value
16         }
17     }
18
19     return true;
20 }
21
22 int main() {
23     char a[1000001];
24     char b[1000001];
25
26
27     scanf("%s", a);
28     scanf("%s", b);
29
30     if (are_compatible(a, b)) {
31         printf("YES\n");
32     } else {
33         printf("NO\n");
34     }
35
36     return 0;
37 }
```

	Input	Expected	Got	
✓	abccba cbbda	YES	YES	✓

Passed all tests! ✓

**Q2)** Danny has a possible list of passwords of Manny's facebook account. All passwords length is odd. But Danny knows that Manny is a big fan of palindromes. So, his password and reverse of his password both should be in the list.

You have to print the length of Manny's password and it's middle character.

**Note:** The solution will be unique.

### INPUT

The first line of input contains the integer N, the number of possible passwords.  
Each of the following N lines contains a single word, its length being an odd number greater than 2 and lesser than **14**. All characters are lowercase letters of the English alphabet.

### OUTPUT

The first and only line of output must contain the length of the correct password and its central letter.

### CONSTRAINTS

**$1 \leq N \leq 100$**

### SAMPLE INPUT

```
4
abc
def
feg
cba
```

### SAMPLE OUTPUT

```
3 b
```

**Solution 2.**  
Denny  
Markus of  
UB  
17 days ago

Denny has a possible list of passwords of Harry's facebook account. All passwords length is odd. But Denny knows that Harry is a big fan of palindromes. So, his password and reverse of his password both should be in the list.

You have to print the length of Harry's password and it's middle character.

Note: The solution will be unique.

#### INPUT

The first line of input contains the integer N, the number of possible passwords.

Each of the following N lines contains a single word, its length being an odd number greater than 2 and lesser than 14. All characters are lowercase letters of the English alphabet.

#### OUTPUT

The first and only line of output must contain the length of the correct password and its central letter.

#### CONSTRAINTS

1 ≤ N ≤ 100

#### SAMPLE INPUT

```
4
abc
cat
dog
tire
```

#### SAMPLE OUTPUT

```
3 b
```

```

1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5
6 void rev(char c[], const char a[]) {
7     int len = strlen(a);
8
9     for (int i = 0; i < len; i++) {
10         c[i] = a[len-i-1];
11     }
12
13     c[len] = '\0';
14 }
15
16 int che(const char a[], const char b[]) {
17     return strcmp(a, b);
18 }
19
20 int main() {
21     int a;
22     scanf("%d", &a);
23
24     char** b = (char**)malloc(a*sizeof(char*));
25
26     for (int i = 0; i < a; i++) {
27         b[i] = (char*)malloc(14*sizeof(char));
28         scanf("%s", b[i]);
29     }
30
31     for (int i = 0; i < a; i++) {
32         char x[14];
33         rev(x, b[i]);
34         for (int j = 0; j < a; j++) {
35             if (i == j)
36                 continue;
37             if (che(x, b[j]) == 0) {
38                 printf("%ld %c", strlen(b[j]), b[j][(strlen(b[j])-1)/2]);
39                 return 0;
40             }
41         }
42     }
43 }

```

	Input	Expected	Got	
✓	4 abc def feg cba	3 b	3 b	✓

Passed all tests! ✓

**Q3)** Joey loves to eat Pizza. But he is worried as the quality of pizza made by most of the restaurants is deteriorating. The last few pizzas ordered 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.

Difficulty: 8  
Domain:  
Data Structures & Algorithms  
C++  
Tags: arrays

Jay-Jay loves to eat Pizza. But he is worried as the quality of pizza made by most of the restaurants is deteriorating. The last few pizzas ordered by him did not taste good. Jay-Jay is feeling extremely hungry and wants to eat pizza. But he is confused about the restaurants from where he should order. So, always he asks Chandler for help.

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

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

#### Input

Read line has N, the total number of restaurants.

Next N lines contain Name of Restaurant and Points awarded by Jay-Jay, 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 Jay-Jay should choose.

#### Constraints

1 <= N <= 10<sup>5</sup>

1 <= Points <= 10<sup>5</sup>

#### SAMPLE INPUT

```
1
Pizzeria 100
Dominos 145
Francesca 85
```

#### SAMPLE OUTPUT

Dominos

#### Explanation

Dominos has maximum points.

```
1 #include <iostream.h>
2 #include <string.h>
3 #include <algorithm>
4
5 typedef struct {
6     char name[21];
7     int points;
8 } Restaurant;
9
10 int compare(const void *a, const void *b) {
11     Restaurant *resta = (Restaurant *)a;
12     Restaurant *restb = (Restaurant *)b;
13
14     if (restA->points < restB->points)
15         return resta->points - restB->points; // Increasing order of points
16     return strcmp(restA->name, restB->name); // lexicographical order of names
17 }
18
19 int main() {
20     int n;
21     cin >> n;
22
23     Restaurant *restaurants = (Restaurant *)malloc(n * sizeof(Restaurant));
24     if (restaurants == NULL) {
25         printf("Memory allocation failed\n");
26         return 1;
27     }
28
29     for (int i = 0; i < n; i++) {
30         string str;
31         cin >> str;
32         restaurants[i].name = str.c_str();
33         cin >> restaurants[i].points;
34     }
35
36     sort(restaurants, restaurants + n, compare);
37
38     cout << restaurants[0].name;
39
40     free(restaurants);
41     return 0;
42 }
```

	Input	Expected	Got
1	1 Pizzeria 100 Dominos 145 Francesca 85	Dominos	Francesca

Passed all tests! ✓

**Q4)** 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 invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him 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**

```
3
1234567890
0123456789
0123456.87
```

**SAMPLE OUTPUT**

```
YES
NO
NO
```

Lesson 4  
DPV  
marked out of  
100  
View discussion

These days, Dr. Chandra is depressed because his crush gave him list of mobile numbers, some of them are valid and some of them are invalid. Dr. Chandra is a special person that he can just like a crack number only if he has valid set of mobile numbers. Help him to determine the valid numbers.

You are given a string "T" and you have to determine whether it's a 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 zeros.

Input:

Get one of input is T representing total number of test cases.  
Next T line each representing "T" as described in the problem statement.

Output:

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

Note: Spaces are not valid.

Constraints:

1 <= T <= 10<sup>5</sup>  
each of string length 1 <= 10<sup>9</sup>

SAMPLE INPUT

```
1
11234567890
01123456789
0112345678T
```

SAMPLE OUTPUT

```
YES
YES
NO
```

```

1. #include <csdio.h>
2. #include <string.h>
3. #include <ctype.h>
4.
5. void check_mobile_number(char s[]) {
6.     if (strlen(s) != 10 || !isdigit(s[0]) || s[0] == '0') {
7.         printf("NO\n");
8.         return;
9.     }
10.    for (int i = 0; i < 10; i++) {
11.        if (!isdigit(s[i])) {
12.            printf("NO\n");
13.            return;
14.        }
15.    }
16.    printf("YES\n");
17. }
18.
19. int main() {
20.     int t;
21.     scanf("%d", &t);
22.     char s[20]; // Assuming input will not exceed 20 characters including '\n' and '\0'
23.
24.     for (int i = 0; i < t; i++) {
25.         scanf("%s", s);
26.         check_mobile_number(s);
27.     }
28.
29.     return 0;
30. }

```

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

Passed all tests! ✓

## WEEK 12

**Q)** 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

- Convert the decimal number 32 to binary number:  $32_{10} = (100000)_2$ .
- The value of the 4th index from the right in the binary representation is 0.

### Sample Case 1

#### Sample Input 1

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.

```
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
8 int fourthBit(int number)
9 {
10     int bin[32];
11     int i=0;
12     while(number>0){
13         bin[i]=number%2;
14         number/=2;
15         i++;
16     }
17     if(i>=4){
18         return bin[3];
19     }
20     else
21         return 0;
22 }
```

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

Passed all tests! ✓

**Q)** 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

10 → n = 10  
3 → p = 3

### **Sample Output 0**

5

### **Explanation 0**

Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3<sup>rd</sup> factor, 5, as the answer.

### **Sample Case 1**

### **Sample Input 1**

STDIN      Function

-----  
10 → n = 10  
5 → 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**

1 → n = 1  
1 → p = 1

### **Sample Output 2**

1

### **Explanation 2**

Factoring n = 1 results in {1}. The p = 1<sup>st</sup> factor of 1 is returned as the 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 m
7 * 2. LONG_INTEGER p
8 */
9
10 long pthFactor(long m, long p)
11 {
12     int c=0;
13     for(long i=1;i<=m;i++){
14         if(m/i==0){
15             c++;
16             if(c==p){
17                 return i;
18             }
19         }
20     }
21     return -1;
22 }
```

	Test	Expected	Got	
✓	printf("1\n", pthFactor(16, 2))	4	4	✓
✓	printf("1\n", pthFactor(16, 5))	#	#	✓
✓	printf("1\n", pthFactor(1, 3))	1	1	✓

Passed all tests! ✓

**Q)** You are a bank account hacker. Initially you have 1 rupee in your account, and you want exactly **N** rupees in your account. You wrote two hacks, first hack can multiply the amount of money you own by 10, while the second can multiply it by 20. These hacks can be used any number of time. Can you achieve the desired amount **N** using these hacks.

### Constraints:

$$1 \leq T \leq 100$$

$$1 \leq N \leq 10^{12}$$

### Input

- The test case contains a single integer N.

### Output

For each test case, print a single line containing the string "1" if you can make exactly N rupees or "0" otherwise.

## SAMPLE INPUT

1

## SAMPLE OUTPUT

1

## SAMPLE INPUT

2

## SAMPLE OUTPUT

0

```
1. /*
2. * complete the 'myFunc' function below.
3. *
4. * The function is expected to return an INTEGER.
5. * The function accepts INTEGER n as parameter.
6. */
7.
8. int myFunc(int n)
9. {
10.     return n>=0 || n==1;
11. }
12.
```

	Test	Expected	Got	
✓	printf("%d", myFunc(1))	1	1	✓
✓	printf("%d", myFunc(2))	0	0	✓
✓	printf("%d", myFunc(10))	1	1	✓
✓	printf("%d", myFunc(25))	0	0	✓
✓	printf("%d", myFunc(200))	1	1	✓

Passed all tests! ✓

**Q)** Find the number of ways that a given integer, **X**, can be expressed as the sum of the **N<sup>th</sup>** powers of unique, natural numbers.

For example, if **X = 13** and **N = 2**, we have to find all combinations of unique squares adding up to **13**. The only solution is  $2^2 + 3^2$ .

### Function Description

Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.

powerSum has the following parameter(s):

X: the integer to sum to

N: the integer power to raise numbers to

Input Format

The first line contains an integer **X**.

The second line contains an integer **N**.

### Constraints

**1 ≤ X ≤ 1000**

**2 ≤ N ≤ 10**

### Output Format

Output a single integer, the number of possible combinations calculated.

### Sample Input 0

10

2

### Sample Output 0

1

### Explanation 0

If **X = 10** and **N = 2**, we need to find the number of ways that **10** can be represented as the sum of squares of unique numbers.

$$\mathbf{10 = 1^2 + 3^2}$$

This is the only way in which **10** can be expressed as the sum of unique squares.

### Sample Input 1

100

2

### Sample Output 1

3

### Explanation 1

$$100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$$

### Sample Input 2

100

3

### Sample Output 2

1

### Explanation 2

**100** can be expressed as the sum of the cubes of **1, 2, 3, 4**.

**(1 + 8 + 27 + 64 = 100)**. There is no other way to express **100** as the sum of cubes.

```
1 //  
2 * Complete the 'powersum' function below.  
3 *  
4 * The function is expected to return an INTEGER.  
5 * The function accepts following parameters:  
6 * 1. INTEGER x  
7 * 2. INTEGER n  
8 */  
9  
10 int powersum(int x, int n, int s)  
11 {  
12     int power = s;  
13     for(int i=1;i<x;i++){  
14         power*=i;  
15     }  
16     if(power > x){  
17         return 0;  
18     }  
19     if(power == x){  
20         return 1;  
21     }  
22     return powersum(x-power,n+1,s) + powersum(x,s+1,n);  
23 }  
24 }
```

Test	Expected	Got
p=int(“100”, powersum(100, 1, 2))	1	1 ✓

Passed all tests! ✓

## WEEK 13

Q1) 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

- $3 \leq n \leq 10^5$
- $1 \leq \text{arr}[i] \leq 2 \times 10^4$ , 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

Explanation 1

- The first and last elements are equal to 1.
- Using zero based indexing,  $\text{arr}[1]=2$  is the pivot between the two subarrays.
- The index of the pivot is 1.

```

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 = 0, right = 0;
11     for(int i=0; i<arr_count; i++){
12         right += arr[i];
13     }
14     for(int i=0; i<arr_count; i++){
15         if(left == (right - arr[i])){
16             return i;
17         }
18         left += arr[i];
19         right -= arr[i];
20     }
21     return 1;
22 }
```

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

Passed all tests! ✓

Q2) 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

$1 \leq n \leq 10^4$

$1 \leq \text{numbers}[i] \leq 10^4$

### **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 numbers.

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

### **Sample Case 0**

#### **Sample Input 0**

STDIN    Function

-----

5    →    numbers[] size n = 5

1    →    numbers = [1, 2, 3, 4, 5]

2

3

4

5

#### **Sample Output 0**

15

#### **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

$12 + 12 = 24$ .

```
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 }  
16
```

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

Passed all tests! ✓

Q3) 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  $|3 - 2| = 1$ .

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

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

Passed all tests! ✓

# Week 14

**Q1)** 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>*, *height<sub>i</sub>***

## 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 40
10 5 41
7 2 42
```

## 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.

Status: Finished  
Started: Sunday, 12 January 2023, 7:37 PM  
Completed: Sunday, 12 January 2023, 7:46 PM  
Duration: 8 mins 50 secs

Question 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 **a**, denoting the number of boxes.

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

#### Constraints

$1 \leq a \leq 100$

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

#### Output Format

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

```
1 #include <stdio.h>
2
3 int main() {
4     int a;
5     scanf("%d", &a);
6
7     for (int i = 0; i < a; i++) {
8         int l, w, h;
9         scanf("%d %d %d", &l, &w, &h);
10
11         if (h < 41) {
12             printf("%d\n", (l*w*h));
13         }
14     }
15 }
```

	Input	Expected	Got	
✓	4 5 5 5 1 2 40 10 5 43 7 2 42	125 0 0 0 0	125 0 0 0 0	✓

Passed all tests! ✓

**Q2)** You are given  $n$  triangles, specifically, their sides  $a_i$ ,  $b_i$  and  $c_i$ . Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides  $a$ ,  $b$  and  $c$  is Heron's formula:

$$S = \sqrt{p * (p - a) * (p - b) * (p - c)} \text{ where } p = (a + b + c) / 2.$$

### Input Format

First line of each test file contains a single integer  $n$ .  $n$  lines follow with  $a_i$ ,  $b_i$  and  $c_i$  on each separated by single spaces.

### Constraints

$$1 \leq n \leq 100$$

$$1 \leq a_i, b_i, c_i \leq 70$$

$$a_i + b_i > c_i, a_i + c_i > b_i \text{ and } b_i + c_i > a_i$$

### Output Format

Print exactly  $n$  lines. On each line print 3 integers separated by single spaces, which are  $a_i$ ,  $b_i$  and  $c_i$  of the corresponding triangle.

### Sample Input 0

```
3
7 24 25
5 12 13
3 4 5
```

### Sample Output 0

```
3 4 5
5 12 13
7 24 25
```

### Explanation 0

The square of the first triangle is **84**. The square of the second triangle is **30**. The square of the third triangle is **6**. So the sorted order is the reverse one.

**Question 2**  
 Contest  
 F. Triangles

You are given  $n$  triangles, specifically, their sides  $a_i$ ,  $b_i$  and  $c_i$ . Print them in the same style but sorted by their areas from the smallest one to the largest one. It is guaranteed that all the areas are different.

The best way to calculate a volume of the triangle with sides  $a$ ,  $b$  and  $c$  is Heron's formula:

$$S = \sqrt{p} * (p - a) * (p - b) * (p - c) \text{ where } p = (a + b + c) / 2.$$

**Input Format:**

First line of each test file contains a single integer  $n$ .  $n$  lines follow with  $a_i$ ,  $b_i$  and  $c_i$  on each separated by single spaces.

**Constraints:**

$1 \leq n \leq 100$   
 $1 \leq a_i, b_i, c_i \leq 70$   
 $a_i + b_i > c_i$ ,  $a_i + c_i > b_i$  and  $b_i + c_i > a_i$ .

**Output Format:**

Print exactly  $n$  lines. On each line print 3 integers separated by single spaces, which are  $a_i$ ,  $b_i$  and  $c_i$  of the corresponding triangle.

```

1 #include <iostream.h>
2 #include <cmath.h>
3 #include <math.h>
4
5 int cap(const void* a, const void* b) {
6     return (*((int*)a) - *((int*)b));
7 }
8
9 int main() {
10     int n;
11     scanf("%d", &n);
12
13     int b[3][3];
14     int c[3], d[3];
15     for (int i = 0; i < n; i++) {
16         scanf("%d %d %d", &b[i][0], &b[i][1], &b[i][2]);
17
18         int p = (b[i][0] + b[i][1] + b[i][2]) / 2;
19         c[i] = 8 * ((p - b[i][0]) * (p - b[i][1]) * (p - b[i][2]));
20         d[i] = sqrt(c[i]);
21     }
22
23     qsort(d, n, sizeof(int), cmp);
24
25     for (int i = 0; i < n; i++) {
26         for (int j = 0; j < n; j++) {
27             if (d[i] == d[j]) {
28                 printf("%d %d %d\n", b[j][0], b[j][1], b[j][2]);
29                 break;
30             }
31         }
32     }
33 }
```

	Input	Expected	Got	
v <sup>2</sup>	1 2 3	1 2 3	v <sup>2</sup>	
	7 24 25	5 12 13	5 12 13	
	5 12 13	7 24 25	7 24 25	
	1 6 5			

Passed all tests! ✓

## WEEK 15

Q) Given an array of integers, reverse the given array in place using an index and loop rather than a built-in function.

### Example

*arr* = [1, 3, 2, 4, 5]

Return the array [5, 4, 2, 3, 1] which is the reverse of the input array.

### Function Description

Complete the function *reverseArray* in the editor below.

*reverseArray* has the following parameter(s):

*int arr[n]*: an array of integers

Return

*int[n]*: the array in reverse order

### Constraints

$1 \leq n \leq 100$

$0 < arr[i] \leq 100$

### Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements in *arr*.

Each line *i* of the *n* subsequent lines (where  $0 \leq i < n$ ) contains an integer, *arr[i]*.

### Sample Case 0

### Sample Input For Custom Testing

5

1

3

2

4

5

### **Sample Output**

5

4

2

3

1

### **Explanation**

The input array is [1, 3, 2, 4, 5], so the reverse of the input array is [5, 4, 2, 3, 1].

### **Sample Case 1**

#### **Sample Input For Custom Testing**

4

17

10

21

45

#### **Sample Output**

45

21

10

17

#### **Explanation**

The input array is [17, 10, 21, 45], so the reverse of the input array is [45, 21, 10, 17].

```

1 //*
2 * Complete the 'reverseArray' function below.
3 *
4 * The function is expected to return an INTEGER_ARRAY.
5 * The function accepts INTEGER_ARRAY arr as parameter.
6 */
7
8 /*
9 * To return the integer array from the function, you should:
10 *     - store the size of the array to be returned in the result_count variable
11 *     - Allocate the array statically or dynamically
12 *
13 * For example,
14 * int* return_integer_array_using_static_allocation(int* result_count) {
15 *     *result_count = 5;
16 *
17 *     static int a[5] = {1, 2, 3, 4, 5};
18 *
19 *     return a;
20 * }
21 *
22 * int* return_integer_array_using_dynamic_allocation(int* result_count) {
23 *     *result_count = 5;
24 *
25 *     int *a = malloc(5 * sizeof(int));
26 *
27 *     for (int i = 0; i < 5; i++) {
28 *         *(a + i) = i + 1;
29 *     }
30 *
31 *     return a;
32 * }
33 */
34
35 int* reverseArray(int arr_count, int *arr, int *result_count) {
36     *result_count=arr_count;
37     for(int i=0;i<arr_count/2;j++)
38     {
39         int temp=arr[i];
40         arr[i]=arr[arr_count-i-1];
41         arr[arr_count-i-1]=temp;
42     }
43     return arr;
44 }
45
46

```

	Test	Expected	Got	
✓	<pre> int arr[] = {1, 3, 2, 4, 5}; int result_count; int* result = reverseArray(5, arr, &amp;result_count); for (int i = 0; i &lt; result_count; i++)     printf("%d\n", *(result + i)); </pre>	5 4 2 3 1	5 4 2 3 1	✓

Q) An automated cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of *minLength* or more, and it can only make one cut at a time. Given the array *lengths[]* representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using this machine. The rod is marked into lengths already, in the order given.

## Example

$n = 3$

$lengths = [4, 3, 2]$

$minLength = 7$

The rod is initially  $sum(lengths) = 4 + 3 + 2 = 9$  units long. First cut off the segment of length  $4 + 3 = 7$  leaving a rod  $9 - 7 = 2$ . Then check that the length 7 rod can be cut into segments of lengths 4 and 3. Since 7 is greater than or equal to  $minLength = 7$ , the final cut can be made.

Return "Possible".

## Example

$n = 3$

$lengths = [4, 2, 3]$

$minLength = 7$

The rod is initially  $sum(lengths) = 4 + 2 + 3 = 9$  units long. In this case, the initial cut can be of length 4 or  $4 + 2 = 6$ . Regardless of the length of the first cut, the remaining piece will be shorter than  $minLength$ . Because  $n - 1 = 2$  cuts cannot be made, the answer is "Impossible".

## Function Description

Complete the function `cutThemAll` in the editor below.

`cutThemAll` has the following parameter(s):

`int lengths[n]:` the lengths of the segments, in order

*int minLength*: the minimum length the machine can accept

Returns

string: "*Possible*" if all  $n-1$  cuts can be made. Otherwise, return the string "*Impossible*".

Constraints

- $2 \leq n \leq 10^5$
- $1 \leq t \leq 10^9$
- $1 \leq lengths[i] \leq 10^9$
- *The sum of the elements of lengths equals the uncut rod length.*

## **Input Format For Custom Testing**

The first line contains an integer,  $n$ , the number of elements in *lengths*.

Each line  $i$  of the  $n$  subsequent lines (where  $0 \leq i < n$ ) contains an integer, *lengths[i]*.

The next line contains an integer, *minLength*, the minimum length accepted by the machine.

## **Sample Case 0**

## **Sample Input For Custom Testing**

STDIN    Function

-----

4 → lengths[] size n = 4

3 → lengths[] = [3, 5, 4, 3]

5

4

3

9 → minLength= 9

## **Sample Output**

Possible

## **Explanation**

The uncut rod is  $3 + 5 + 4 + 3 = 15$  units long. Cut the rod into lengths of  $3 + 5 + 4 = 12$  and 3. Then cut the 12 unit piece into lengths 3 and  $5 + 4 = 9$ . The remaining segment is  $5 + 4 = 9$  units and that is long enough to make the final cut.

## **Sample Case 1**

### **Sample Input For Custom Testing**

STDIN    Function

-----

3 → lengths[] size n = 3

5 → lengths[] = [5, 6, 2]

6

2

12 → minLength= 12

## **Sample Output**

Impossible

## **Explanation**

The uncut rod is  $5 + 6 + 2 = 13$  units long. After making either cut, the rod will be too short to make the second cut.

```

3 /**
4  * Completes the 'cutThemAll' function below.
5  *
6  * The function is expected to return a STRING.
7  * The function accepts following parameters:
8  * 1. LONG_INTEGER_ARRAY lengths
9  * 2. LONG_INTEGER minLength
10 */
11
12 /**
13  * To return the string from the function, you should either do static allocation or dynamic allocation
14  *
15  * For example,
16  * char* return_string_using_static_allocation() {
17  *     static char s[] = "static allocation of string";
18  *
19  *     return s;
20  * }
21
22 * char* return_string_using_dynamic_allocation() {
23 *     char* s = malloc(100 * sizeof(char));
24 *
25 *     s = "dynamic allocation of string";
26 *
27 *
28 */
29 char* cutThemAll(int lengths_count, long *lengths, long minLength) {
30     long t=0,i=1;
31     for(int i=0;i<lengths_count-1;i++)
32     {
33         t+=lengths[i];
34     }
35     do
36     {
37         if(t-lengths[lengths_count-1]<minLength)
38         {
39             return "Impossible";
40         }
41         i++;
42     }while(i<lengths_count-1);
43     return "Possible";
44 }

```

	Test	Expected	Got	
✓	long lengths[] = {3, 5, 4, 3}; printf("%s", cutThemAll(4, lengths, 9))	Possible	Possible	✓
✓	long lengths[] = {5, 6, 2}; printf("%s", cutThemAll(3, lengths, 12))	Impossible	Impossible	✓

Passed all tests! ✓