



Week - 1



Practice problems for week - 1 (Week of 14 August)

Question 1

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Question 2

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Question 3

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Question 4

Problem Description

Input format

Output Format

Sample input and output

Solution

Question 5

Problem Description

Input constraints

Input format

Output Format

Sample input and output

Solution

Practice problems for week - 1 (Week of 14 August)

None of the following problems require a knowledge of loops, conditional statements etc.

Question 1

Problem Description

Keval has forgotten the password to his I-Pad. The password is a **4-digit** number where each digit takes a value from **0 to 9**. The good thing is that Keval remembers that his password had exactly two unique digits, and each of these digits appeared exactly twice in the password. Keval also remembers that n digits (where, $0 \leq n \leq 10$) from 0-9 were definitely not used in the password.

Find the number of different possible passwords Keval could have.

Note that the password can start with the digit 0

Input constraints

$$0 \leq n \leq 10$$

Input format

The only line of input contains a single integer n , the number of digits which were not used in the password

Output Format

Output a single integer that denotes the number of possible password sequences.

Sample input and output

Sample Input	Sample Output
1	216

Solution

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

```
int main() {  
    int n;  
    scanf("%d", &n);  
    int ans = ((10-n)*(10-n-1))/2 * 6; // (10-n)C2 * 4C2  
    printf("%d\n", ans);  
    return 0;  
}
```

Question 2

Problem Description

You are given two integers x and y . Output the minimum integer z , such that $x \mid z = y$. Here, " \mid " denotes the [bitwise-OR operator](#)

Input constraints

$$1 \leq x, y \leq 10^9$$

Input format

The only line of input contains two space-separated integers x and y

Output Format

Output a single integer that denotes z as described in the problem statement

Sample input and output

Sample Input	Sample Output
1 3	2

Solution

```
#include <stdio.h>

int main() {
    int x, y;
    scanf("%d %d", &x, &y);

    int z = x ^ y;
    printf("%d", z);

    return 0;
}
```

Question 3

Problem Description

Reverse a 4-digit number using the modulo operator.

Input constraints

The input would be a valid 4-digit number (i.e. no leading zeroes would be present)

Input format

The only line of input contains a single integer n

Output Format

Output a single integer which denotes the reverse of the n . Note that the output may have leading zeroes.

Sample input and output

Sample Input	Sample Output
3214	4123
5600	0065

Solution

```
#include <stdio.h>

int main() {
    int n, reversed = 0;

    // Input a 4-digit number
    printf("Enter a 4-digit number: ");
    scanf("%d", &n);

    // Reverse the number using modulo operator
    reversed = (n % 10) * 1000 + ((n / 10) % 10) * 100 + ((n / 100) % 10) * 10 + (n / 1000);

    // Output the reversed number
    printf("Reversed number: %d\n", reversed);
}
```

```
    return 0;  
}
```

Question 4

Problem Description

Given a time as input in `hh:mm:ss` format, print the total number of minutes which have passed in the day.

Input format

The only line of input would contain the time in `hh:mm:ss` format (24-hour clock)

Output Format

Output the total number of minutes that have passed. Note that this may not be an integer.

Sample input and output

Sample Input	Sample Output
13:40:30	820.5

Solution

```
#include <stdio.h>  
  
int main() {  
    // Declare three variables that will be used to store the hours, minutes,  
    // and seconds, respectively
```

```

int h, m, s;

// Take input in the specified format
scanf("%d:%d:%d", &h, &m, &s);

/**
 * Calculate the total minutes
 * Total minutes = hours * 60 + minutes + seconds / 60
 * However, to ensure that float division is done, we typecast the seconds
 * to float
 */
float minutes = 60 * h + m + (float)s / 60;

// Output the total minutes as a floating point number
printf("%f", minutes);
}

```

Question 5

Problem Description

Kishore is a very strange boy. He keeps solving weird problems and is now stuck on one such question. Please help him solve it.

Given an integer n , you need to find the number of **ordered** pairs of integers (a, b) such that, $1 \leq a \leq b \leq n$ and

$$\frac{lcm(a,b)}{gcd(a,b)} \leq 3$$

In this problem, $gcd(a, b)$ denotes the [greatest common divisor](#) of the numbers a and b , and $lcm(a, b)$ denotes their [lowest common multiple](#).

Input constraints

$$1 \leq n \leq 10^8$$

Input format

The only line of input contains a single integer n

Output Format

Output a single integer that denotes the number of pairs (a, b) that satisfy the given constraints

Sample input and output

Sample Input	Sample Output
5	11

Solution

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

```
int main() {  
    int n;  
    scanf("%d", &n);  
  
    int ans = n + 2 * ((n / 2) + (n / 3));  
    printf("%d", n);  
  
    return 0;  
}
```

