

Question **1**

Correct

Marked out of 1.00

 Flag question

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

#### 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

STDIN Function

-----

77 → number = 77

#### Sample Output 1

1

#### Explanation 1

- Convert the decimal number 77 to binary number:  $77_{10} = (1001101)_2$ .

Question **4**

Correct

Marked out of 1.00

 Flag question

**Input Format**

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

**Output Format**

- In the first line print two space-separated integers, representing the length of *a* and *b* respectively.
- In the second line print the string produced by concatenating *a* and *b* (*a + b*).
- In the third line print two strings separated by a space, *a'* and *b'*. *a'* and *b'* are the same as *a* and *b*, respectively, except that their first characters are swapped.

**Sample Input**

abcd  
ef

**Sample Output**

4 2  
abcdef  
ebcd af

**Explanation**

a = "abcd"  
b = "ef"  
|a| = 4  
|b| = 2  
a + b = "abcdef"  
a' = "ebcd"  
b' = "af"

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     char str1[10],str2[10],t;
4     int i=0,j=0;
5     int c1=0,c2=0;
6     scanf("%s",str1);
7     scanf("%s",str2);
8     while(str1[i]!='\0'){
9         c1++;
10        i++;
11    }
12    while(str2[j]!='\0'){
13        c2++;
14        j++;
15    }
16    printf("Xd Xd\n",c1,c2);
17    printf("XsXs\n",str1,str2);
18    t=str1[0];
19    str1[0]=str2[0];
20    str2[0]=t;
21    printf("Xs Xs",str1,str2);
22    return 0;
23 }
24
25
```

	Input	Expected	Got	
✓	abcd ef	4 2 abcdef ebcd af	4 2 abcdef ebcd af	✓

Passed all tests! ✓

Question **1**

Correct

Marked out of 1.00

Flag question

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

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

**Constraints:**

- 1 <= t <= 10***
- 1 <= n <= 1000***
- 1 <= a[i] <= 1000***

**Input:**

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

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

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

**Output:**

For every test case print your answer in new line

SAMPLE INPUT

1  
5 1  
1 2 3 4 5

SAMPLE OUTPUT

4

Explanation

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓