Question **1**

Correct

Marked out of 1.00

Flag question

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<=T<=100

1<=N<=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

7:43 PM

0

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an
 * The function accepts INTEGER n as para
 */

int myFunc(int n)
{
    while (n>1) {
        if(n%20==0){
            n/=20;
        }else if(n%10==0){
            n/=10;
        }else{
            return 0;
        }
    }
    return n==1?1:0;
}

void start(){
     int T,N;
     scanf("%d",&T);
     while(T--){
         scanf("%d", &N);
         printf("%d",myFunc(N));
     }
}
```

	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! ✓

estion **2**

rrect

arked out of
00

Flag question

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

7:43 PM

Question **2**

Correct

Marked out of 1.00

Flag question

Find the number of ways that a given integer, ***X***, can be expressed as the sum of the N^{th} 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

$1 \le X \le 1000$

$2 \le N \le 10$

Output Format

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

Sample Input 0

```
10
2
```

Sample Output 0

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.

$10 = 1^2 + 3^2$

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

7:43 PM

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.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'powerSum' function below
 *
 * The function is expected to return an
 * The function accepts following paramet
 *  1. INTEGER x
 *  2. INTEGER n
 */

int powerSum(int x, int m, int n)
{
    if(x==0)
    {
      return 1;
    }
    if(x<0)
    {
      return 0;
    }
    int count =0;
    for(int i=m;;i++)
    {
        int power=1;
        for(int j=0;j<n;j++)
        {
        power*=i;
        }
        if(power>x)
        {
        break;
        }
        count+=powerSum(x-power,i+1,n);
    }
return count;
}
```

est	Expected	Got	
rintf("%d", powerSum(10, 1, 2))	1	1	✓

Passed all tests! ✓

Finish review

7:43 PM

Week-13-Pas...
lakshmicolleges.org

Completed	Monday, 13 January 2025, 8:40 PM
Duration	10 mins 38 secs

Question 1
Correct
Flag question

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

Example

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

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

Function Description

Complete the function balancedSum in the editor below.

balancedSum has the following parameter(s):

int arr[n]: an array of integers

Returns:

int: an integer representing the index of the pivot

Constraints

- $3 \le n \le 10^5$
- $1 \le arr[i] \le 2 \times 10^4$, where $0 \le i < n$
- 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 ≤ i

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 ≤ 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, 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

7:44 PM

Week-13-Pas...
lakshmicolleges.org

Explanation 1

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

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'balancedSum' function be
 *
 * The function is expected to return an
 * The function accepts INTEGER_ARRAY arr
 */

int balancedSum(int arr_count, int* arr)
{
    int totalsum=0;
    for(int i=0;i<arr_count;i++){
        totalsum+=arr[i];
    }
    int leftsum=0;
    for(int i=0;i<arr_count;i++){
        int rightsum=totalsum-leftsum-arr
        if(leftsum==rightsum){
            return i;
        }
        leftsum+=arr[i];
    }
    return 1;
}
```

	Test	Expected

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

Passed all tests! ✓

Question **2**
Correct
Flag question

Calculate the sum of an array of integers.

Example

7:44 PM

Passed all tests! ✓

Question **2**

Correct

Flag question

Calculate the sum of an array of integers.

Example

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

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

Function Description

Complete the function arraySum in the editor below.

arraySum has the following parameter(s):

int numbers[n]: an array of integers

Returns

int: integer sum of the numbers array

Constraints

$1 \le n \le 10^4$

$1 \le numbers[i] \le 10^4$

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 $0 \le i < n$.

Sample Case 0

Sample Input 0

```
STDIN    Function
-----    --------
5     →  numbers[] size n = 5
```

```
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.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'arraySum' function below
 *
 * The function is expected to return an
 * The function accepts INTEGER_ARRAY num
 */

int arraySum(int numbers_count, int *numb
```

12 → numbers = [12, 12]

12

Sample Output 1

24

Explanation 1

12 + 12 = 24.

Answer: (penalty regime: 0 %)

Reset answer

```
 Complete the 'arraySum' function below.

 The function is expected to return an INT
 The function accepts INTEGER_ARRAY number
/


t arraySum(int numbers_count, int *numbers

  int sum=0;
  for(int i=0;i<numbers_count;i++){
      sum=sum+numbers[i];
  }
  return sum;
```

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

Passed all tests! ✓

Question **3**
Correct
Flag question

Given an array of n integers, rearrange them so that the sum of the absolute differences of all adjacent elements is minimized. Then, compute the sum of those absolute differences. Example n = 5 arr = [1, 3, 3, 2, 4] If the list is rearranged as arr' = [1, 2, 3, 3, 4], the absolute differences are |1 - 2| = 1, |2 - 3| = 1, |3 - 3| = 0, |3 - 4| = 1. The sum of those differences is 1 + 1 + 0 + 1 = 3. Function Description

7:44 PM

Question **3**

Correct

Flag question

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

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an
 * The function accepts INTEGER_ARRAY arr
 */
#include <stdlib.h>
int compare(const void *a,const void *b){
    return (*(int *)a-*(int *)b);
}
int minDiff(int arr_count, int* arr)
{
    qsort(arr,arr_count,sizeof(int),compa
    int totaldiff = 0;
    for(int i=1;i<arr_count;i++){
        totaldiff+=abs(arr[i]-arr[i-1]);
    }
    return totaldiff;
}

```

	Test	Expected	Got
✓	int arr[] = {5, 1, 3, 7, 3};	6	6

7:44 PM

|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 ≤ n ≤105 0 ≤ arr[i] ≤ 109, where 0 ≤ 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 ≤ i < n) . Sample Case 0 Sample Input For Custom Testing STDIN Function ----- -------- 5 → arr[] size n = 5 5 → arr[] = [5, 1, 3, 7, 3] 1 3 7 3 Sample Output 6 Explanation n = 5 arr = [5, 1, 3, 7, 3] If arr is rearranged as arr' = [1, 3, 3, 5, 7], the differences are minimized. The final answer is |1 - 3| + |3 - 3| + |3 - 5| + |5 - 7| = 6. Sample Case 1 Sample Input For Custom Testing STDIN Function ----- -------- 2 → arr[] size n = 2 3 → arr[] = [3, 2] 2 Sample Output 1 Explanation n = 2 arr = [3, 2] There is no need to rearrange because there are only two elements. The final answer is |3 - 2| = 1.

Answer: (penalty regime: 0 %)

Reset answer

```
/*
 * Complete the 'minDiff' function below.
 *
 * The function is expected to return an
 * The function accepts INTEGER_ARRAY arr
 */
#include <stdlib.h>
int compare(const void *a,const void *b){
    return (*(int *)a-*(int *)b);
}
int minDiff(int arr_count, int* arr)
{
    qsort(arr,arr_count,sizeof(int),compa
    int totaldiff = 0;
    for(int i=1;i<arr_count;i++){
        totaldiff+=abs(arr[i]-arr[i-1]);
    }
    return totaldiff;
}

```

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

Passed all tests! ✓

Finish review