Sample Case 1

Sample Input

1288

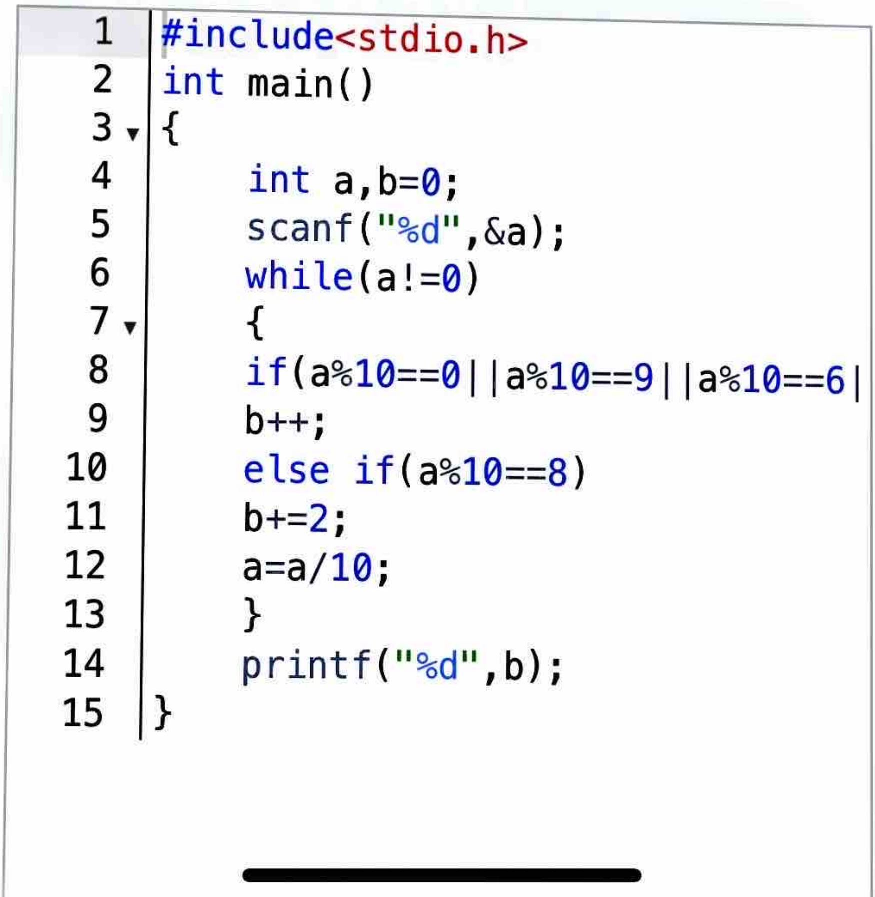
Sample Output

4

Explanation

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

Answer: (penalty regime: 0 %)



```
#include<stdio.h>
int main()
{
    int a,b=0;
    scanf("%d",&a);
    while(a!=0)
    {
    if(a%10==0||a%10==9||a%10==6|
    b++;
    else if(a%10==8)
    b+=2;
    a=a/10;
    }
    printf("%d",b);
}
```

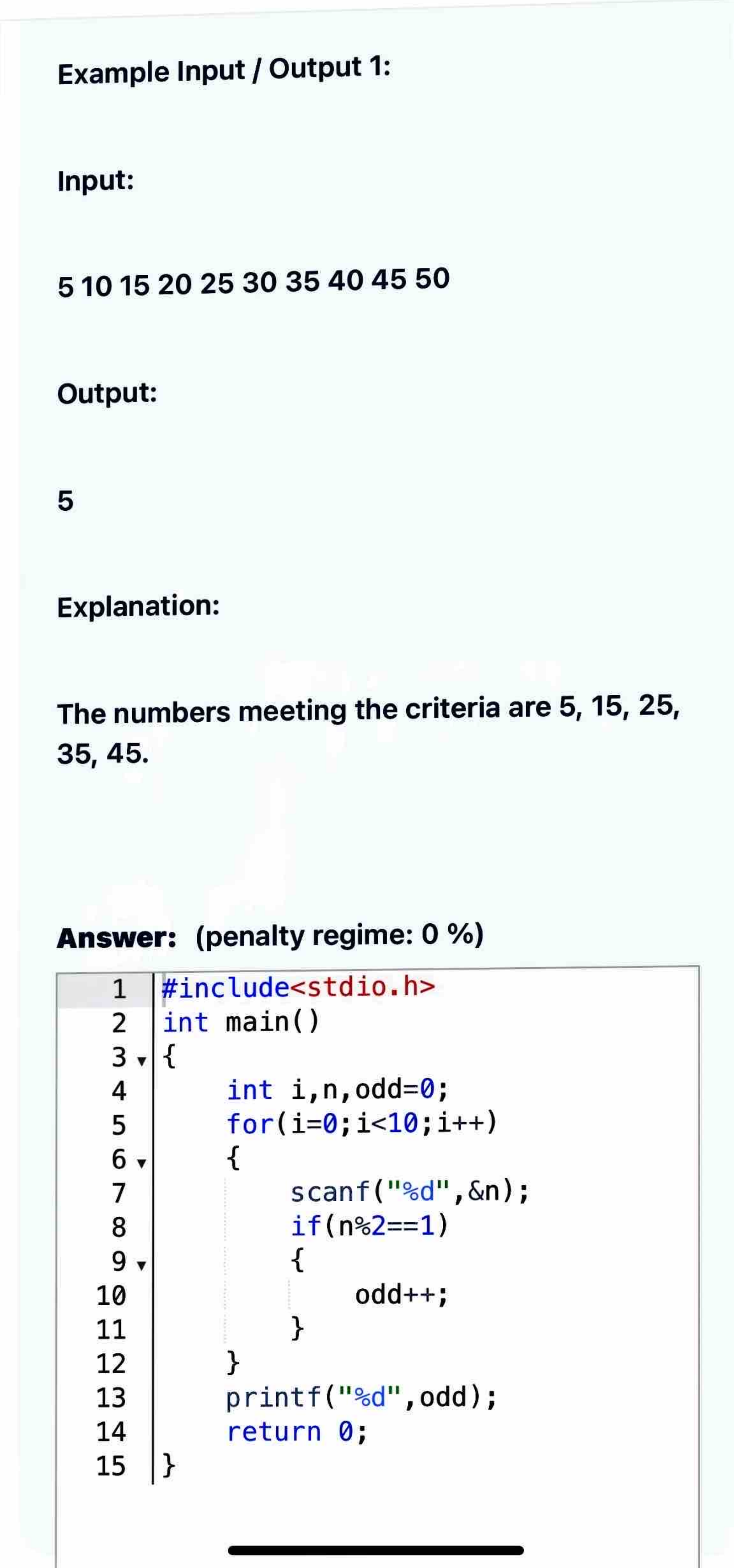
are enough to purchase any item ranging from $1 to $10. Hence minimum is 4. Likewise denominations could also be {$1, $2, $3, $5}. Hence answer is still 4.

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 %)



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 %)

```
#include<stdio.h>
int main()
{
    int i,n,odd=0;
    for(i=0;i<10;i++)
    {
        scanf("%d",&n);
        if(n%2==1)
        {
            odd++;
        }
    }
    printf("%d",odd);
    return 0;
}
```

Output: false

Explanation:

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

Note:

1. 0 <= N <= 10^9

2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
        int n,rem,rev=0;
        scanf("%d",&n);
        rem=n%10;
        if(rem==0||rem==1||rem==6||re
        {
                while(n!=0)
                {
                        rem=n%1;
                        rev=rev*10+rem;
                        n=n/10;
                }
                printf("true");
        }
        else
        {
                printf("false");
        }
        return 0;
}
```

Sample Input For Custom Testing

Sample Input 2

```
3
3
```

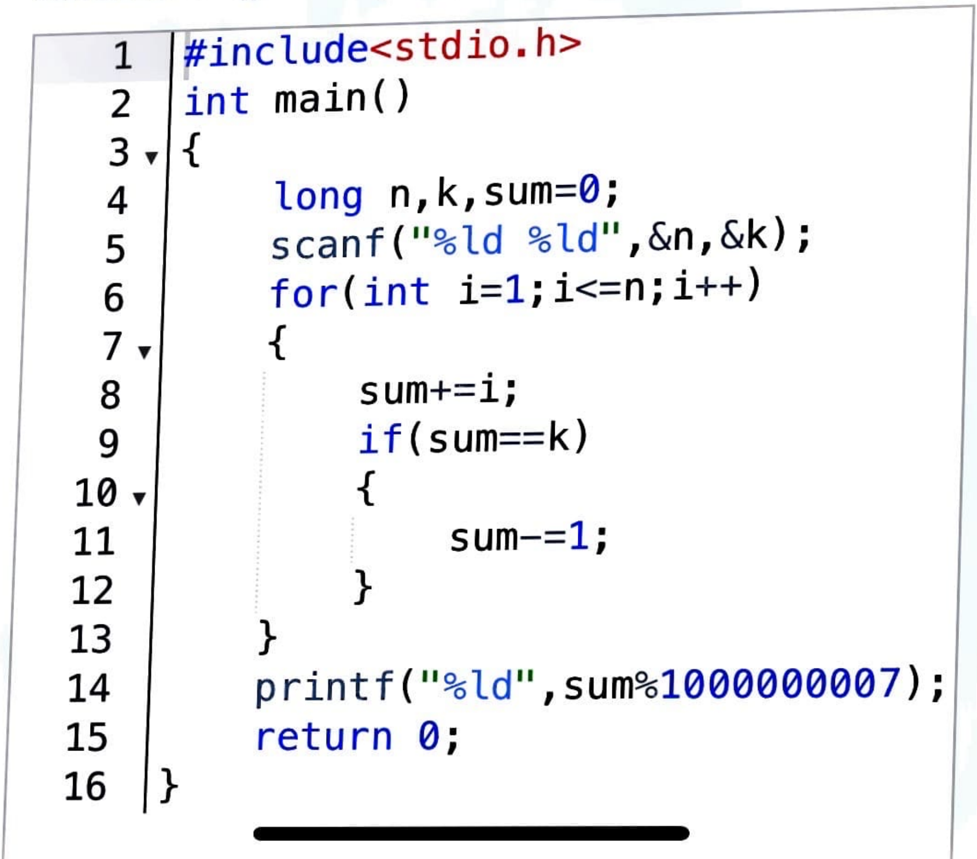
Sample Output 2

```
5
```

Explanation 2

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

Answer: (penalty regime: 0 %)



```c
#include<stdio.h>
int main()
{
        long n,k,sum=0;
        scanf("%ld %ld",&n,&k);
        for(int i=1;i<=n;i++)
        {
                sum+=i;
                if(sum==k)
                {
                        sum-=1;
                }
        }
        printf("%ld",sum%1000000007);
        return 0;
}
```