

Rajalakshmi Engineering College

Name: Tejeshwaran P

Email: 241801292@rajalakshmi.edu.in

Roll no: 241801292

Phone: 6383048813

Branch: REC

Department: I AI & DS AF

Batch: 2028

Degree: B.E - AI & DS

Scan to verify results



NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 4_PAH

Attempt : 2

Total Mark : 50

Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Amar is working on a project where he needs to implement a special type of queue that allows selective dequeuing based on a given multiple. He wants to efficiently manage a queue of integers such that only elements not divisible by a given multiple are retained in the queue after a selective dequeue operation.

Implement a program to assist Amar in managing his selective queue.

Example

Input:

5

Output: 10 2 30 4 50

Original

Queue: 10 2

30 4 50 Queue after selective dequeue: 2 4

Explanation:

After selective dequeue with a multiple of 5, the elements that are multiples of 5 should be removed. Therefore, only 10, 30, and 50 should be removed from the queue. The updated Queue is 2 4.

Input Format

The first line contains an integer n , representing the number of elements initially present in the queue.

The second line contains n space-separated integers, representing the elements of the queue.

The third line contains an integer multiple, representing the divisor for selective dequeue operation.

Output Format

The first line of output prints "Original Queue: " followed by the space-separated elements in the queue before the dequeue operation.

The second line prints "Queue after selective dequeue: " followed by the remaining space-separated elements in the queue, after deleting elements that are the multiples of the specified number.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

10 2 30 4 50

5

Output: Original

Queue: 10 2 30 4 50

Queue after selective
dequeue: 2 4

Answer

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

```
#define MAX 50
```

```
int main() {    int n, i, multiple;
```

```
    int queue[MAX], filteredQueue[MAX];
```

```
    int filteredSize = 0;
```

```
    scanf("%d", &n);    for (i = 0; i < n; i++) {
```

```
        scanf("%d", &queue[i]);
```

```
    }
```

```
    scanf("%d", &multiple);
```

```
    printf("Original Queue:");    for (i = 0; i < n; i++) {
```

```
    printf(" %d", queue[i]);    if (queue[i] % multiple  
    != 0) {
```

```
        filteredQueue[filteredSize++] = queue[i];
```

```
    }
```

```
    }
```

```
    printf("\nQueue after selective dequeue:");
```

```
    for (i = 0; i < filteredSize; i++) {
```

```
        printf(" %d", filteredQueue[i]);
```

```
    }
```

```
    printf("\n");
```

```
    return 0; }
```

Status : Correct

Marks : 10/10

]