

9/11/20

DS lab-test -1

Anisudha Acharya

BM19CS193

- ③ Given a queue of whole numbers. WAP to return a new queue which contains the evenly divisible nos. Assume that there will always be a few elements in the input queue, which are evenly divisible by the nos in the mentioned range.

Note: A no. is said to be evenly divisible if it is divisible by all nos in the given range without any remainder. Consider the range to be from 1 to 10 (both inclusive)

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

```
#define MAX 5
```

```
int q[MAX]
```

```
int p[MAX]
```

```
int rear1 = -1;
```

```
int front1 = -1;
```

```
int rear2 = -1;
```

```
int front2 = -1;
```

```
int i;
```

```
void enqueue1(int ele)
```

```
{
```

```
    rear1 = (rear1 + 1) % MAX;
```

```
    p[rear2] = ele;
```

```
    printf("%d enqueued in p\n", ele);
```

```
    if (front2 == -1)
```

```
        front2 = 0;
```

```
}
```

```
void display()
```

```
{
```

```
    print("The elements divisible by all no.s from 1 to 10 are :");
```

```
    for (i = front; i <= rear; i++)
```

```
    {
        printf("%d\t", p[i]);
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int item;
```

```
    printf("Enter the queue elements");
```

```
    for (i = 0; i < MAX; i++)
```

```
    {
        scanf("%d", &item);
```

```
        enqueue(item);
```

```
    }
```

```
    for (i = front; i <= rear; i++)
```

```
    {
```

```
        if ((a[i] % 2 == 0) && a[i] % 3 == 0 && a[i] % 4 == 0)
```

```
            && (a[i] % 5 == 0) && (a[i] % 6 == 0) && (a[i] % 7 == 0)
```

```
            && (a[i] % 8 == 0) && (a[i] % 9 == 0) && (a[i] % 10 == 0)
```

```
        {
            enqueue 2(a[i]);
```

```
        }
```

```
    }
```

```
}
```

```
    display();
```

```
    return 0;
```

```
}
```