


NAME – GAGAN VIJAY BHANGALE

1)

```
import java.io.*;
import java.util.Scanner;
class Lab
{
    static String isDivisible(long n)
    {
        long temp = n;
        int sum = 0;
        while (n != 0)
        {
            int k = (int) n % 10;
            sum += k;
            n /= 10;
        }
        if (temp % sum == 0)
            return "YES it is divisible";
        return "NO it is not divisible";
    }
}
```

```
}  
  
public static void main(String []args)  
{  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter Your Number = ");  
    long n =sc.nextLong();  
    System.out.println(isDivisible(n));  
}  
}
```



```
Enter Your Number =  
40  
YES it is divisible  
|
```

```

Lab.java ×
1 import java.io.*;
2 import java.util.Scanner;
3 class Lab
4 {
5     static String isDivisible(long n)
6     {
7         long temp = n;
8
9         int sum = 0;
10        while (n != 0)
11        {
12            int k = (int) n % 10;
13            sum += k;
14            n /= 10;
15        }
16
17        if (temp % sum == 0)
18            return "YES it is divisible";
19
20        return "NO it is not divisible";
21    }
22
23    public static void main(String []args)
24    {
25        Scanner sc = new Scanner(System.in);
26        System.out.println("Enter Your Number = ");
27        long n = sc.nextLong();
28        System.out.println(isDivisible(n));
29    }
30 }
31
32

```

Problems @ Javadoc Declaration Console ×

```

<terminated> Lab [Java Application] C:\Users\Admin\Downloads\sts-4.17.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
Enter Your Number =
40
YES it is divisible

```

2)

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

pthread_mutex_t mutex;

void *thread_function(void *arg)
{
    // Acquire a lock before printing data
    pthread_mutex_lock(&mutex);

    printf("Thread Function: %s\n", (char *)arg);

    // Release the lock
    pthread_mutex_unlock(&mutex);

    return NULL;
}

int main()
{
    pthread_t thread1, thread2, thread3;
    char *msg1 = "Thread 1";
    char *msg2 = "Thread 2";
    char *msg3 = "Thread 3";

    // Initialize the mutex
    pthread_mutex_init(&mutex, NULL);

    // Create the threads
    pthread_create(&thread1, NULL, thread_function, (void *)msg1);
    pthread_create(&thread2, NULL, thread_function, (void *)msg2);
    pthread_create(&thread3, NULL, thread_function, (void *)msg3);

    // Join the threads
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);
    pthread_join(thread3, NULL);

    // Destroy the mutex
    pthread_mutex_destroy(&mutex);

    return 0;
}
```

day5 > C thread.c > thread_function(void *)

```
1  #include <stdio.h>
2  #include <pthread.h>
3
4  pthread_mutex_t mutex;
5  void *thread_function(void *arg)
6  {
7      // Acquire a lock before printing data
8      pthread_mutex_lock(&mutex);
9
10     printf("Thread Function: %s\n", (char *)arg);
11
12     // Release the lock
13     pthread_mutex_unlock(&mutex);
14     return NULL;
15 }
16
17 int main()
18 {
19     pthread_t thread1, thread2, thread3;
20     char *msg1 = "Thread 1";
21     char *msg2 = "Thread 2";
22     char *msg3 = "Thread 3";
23
24     // Initialize the mutex
25     pthread_mutex_init(&mutex, NULL);
26
27     // Create the threads
28     pthread_create(&thread1, NULL, thread_function, (void *)msg1);
29     pthread_create(&thread2, NULL, thread_function, (void *)msg2);
30     pthread_create(&thread3, NULL, thread_function, (void
31 *)msg3);
32
33     // Join the threads
34     pthread_join(thread1, NULL);
35     pthread_join(thread2, NULL);
36     pthread_join(thread3, NULL);
37
38     // Destroy the mutex
39     pthread_mutex_destroy(&mutex);
```

```
day5 > C thread.c > thread_function(void *)
14 |     return NULL;
15 | }
16 |
17 | int main()
18 | {
19 |     pthread_t thread1, thread2, thread3;
20 |     char *msg1 = "Thread 1";
21 |     char *msg2 = "Thread 2";
22 |     char *msg3 = "Thread 3";
23 |
24 |     // Initialize the mutex
25 |     pthread_mutex_init(&mutex, NULL);
26 |
27 |     // Create the threads
28 |     pthread_create(&thread1, NULL, thread_function, (void *)msg1);
29 |     pthread_create(&thread2, NULL, thread_function, (void *)msg2);
30 |     pthread_create(&thread3, NULL, thread_function, (void
31 | *)msg3);
32 |
33 |     // Join the threads
34 |     pthread_join(thread1, NULL);
35 |     pthread_join(thread2, NULL);
36 |     pthread_join(thread3, NULL);
37 |
38 |     // Destroy the mutex
39 |     pthread_mutex_destroy(&mutex);
40 |
41 |     return 0;
42 | }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Thread Function: Thread 2
osboxes@osboxes:~/one/day5\$ cd "/home/osboxes/one/day5/" && gcc thread.c -o thread && "/home/osboxes/one/day5/"thread
Thread Function: Thread 3
Thread Function: Thread 2
Thread Function: Thread 1
osboxes@osboxes:~/one/day5\$

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - day5 + -

```
cd "/home/osboxes/one/day5/" && gcc thread.c -o thread && "/home/osboxes/one/day5/"thread
osboxes@osboxes:~/one$ cd "/home/osboxes/one/day5/" && gcc thread.c -o thread && "/home/osboxes/one/day5/"th
read
Thread Function: Thread 3
Thread Function: Thread 1
Thread Function: Thread 2
osboxes@osboxes:~/one/day5$
```