**Ahmed Hassan Ismail – BESE-9B – 237897**

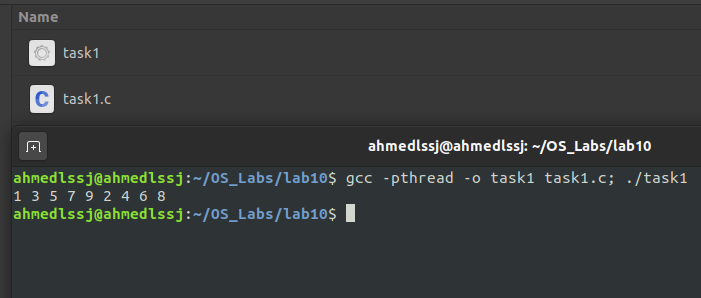
**OS Lab 10**

**Task1**

**Code:**

Given in Lab manual.

**Screenshot:**



**Task 2**

**Code:**

#include "stdio.h"

#include "stdlib.h"

#include "pthread.h"

pthread\_mutex\_t count\_mutex = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t condition\_var = PTHREAD\_COND\_INITIALIZER;

int count = 0;

#define COUNT\_DONE 10

// Print odd numbers

void \*printOddNums()

{

    for (;;)

    {

        // Lock mutex and then wait for signal to relase mutex

        pthread\_mutex\_lock(&count\_mutex);

        if (count % 2 != 0)

        {

            pthread\_cond\_wait(&condition\_var, &count\_mutex);

        }

        count++;

        printf("%d ", count);

        pthread\_cond\_signal(&condition\_var);

        if (count >= COUNT\_DONE)

        {

            pthread\_mutex\_unlock(&count\_mutex);

            return (NULL);

        }

        pthread\_mutex\_unlock(&count\_mutex);

    }

}

// print even numbers

void \*printEvenNums()

{

    for (;;)

    {

        // Lock mutex and then wait for signal to relase mutex

        pthread\_mutex\_lock(&count\_mutex);

        if (count % 2 == 0)

        {

            pthread\_cond\_wait(&condition\_var, &count\_mutex);

        }

        count++;

        printf("%d ", count);

        pthread\_cond\_signal(&condition\_var);

        if (count >= COUNT\_DONE)

        {

            pthread\_mutex\_unlock(&count\_mutex);

            return (NULL);

        }

        pthread\_mutex\_unlock(&count\_mutex);

    }

}

int main()

{

    pthread\_t thread1, thread2;

    pthread\_create(&thread1, NULL, &printOddNums, NULL);

    pthread\_create(&thread2, NULL, &printEvenNums, NULL);

    pthread\_join(thread1, NULL);

    pthread\_join(thread2, NULL);

    printf("\n");

    exit(0);

}

**Screenshot:**

