

LAB 8

Q1)Write a multithread program that generates the Fibonacci Series:

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

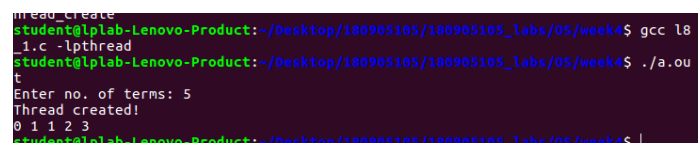
void *thread_code(void* param)
{
    int n = *((int*)param);
    int arr[n];
    arr[0] = 0;
    arr[1] = 1;
    for (int i = 2; i < n; i++)
        arr[i] = arr[i - 2] + arr[i - 1];
    int *sol = (int *)calloc(n, sizeof(int));
    memcpy(sol, arr, sizeof(int) * n);
    return sol;
}

void main()
{
    int n;
    printf("Enter no. of terms: ");
    scanf("%d", &n);
    void *arr;

    pthread_t thread;
    pthread_create(&thread, 0, &thread_code, (void*)&n);
    printf("Thread created!\n");
    pthread_join(thread, &arr);

    int *arr2 = arr;
    for (int i = 0; i < n; i++)
        printf("%d ", (int)arr2[i]);
    printf("\n");
}
```

output:



```
inread_create
student@lplab-Lenovo-Product:~/Desktop/180905105/180905105_labs/OS/week4$ gcc l8
_1.c -lpthread
student@lplab-Lenovo-Product:~/Desktop/180905105/180905105_labs/OS/week4$ ./a.out
Enter no. of terms: 5
Thread created!
0 1 1 2 3
student@lplab-Lenovo-Product:~/Desktop/180905105/180905105_labs/OS/week4$ |
```

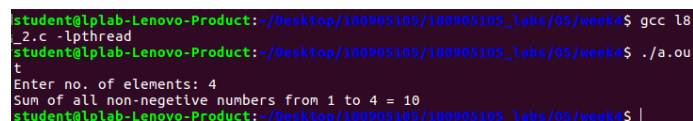
Q2)Write a multithread program that calculates the summation of non-negative integers

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

void* thread_code(void* n)
{
    int sum = 0;
    int limit = *((int*)n);
    for (int i = 1; i <= limit; i++)
        sum += i;
    *((int*)n) = sum;
}

void main()
{
    int n, num;
    printf("Enter no. of elements: ");
    scanf("%d", &n);
    num = n;
    pthread_t thread;
    pthread_create(&thread, 0, &thread_code, (void*)&n);
    pthread_join(thread, NULL);
    printf("Sum of all non-negative numbers from 1 to %d = %d\n", num, n);
}
```

OUTPUT:



```
student@iplab-Lenovo-Product:~/desktop/180905105/180905105_labs/05/week4$ gcc l8_2.c -lpthread
student@iplab-Lenovo-Product:~/desktop/180905105/180905105_labs/05/week4$ ./a.out
Enter no. of elements: 4
Sum of all non-negative numbers from 1 to 4 = 10
student@iplab-Lenovo-Product:~/desktop/180905105/180905105_labs/05/week4$ |
```

Q3)Write a multithread program for generating prime number from a given starting number to a given ending number

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

void *prime(void* arr)
{
    int n1 = *((int*)arr);
    int n2 = (((int*)(arr + sizeof(int)))));
    printf("Prime numbers: ");
    for (int i = n1; i <= n2; i++)
    {
        int flag = 0;
```


```

        for (int j = 2; j <= i / 2; j++)
        {
            if ((i % j) == 0)
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            printf("%d ", i);
    }
    printf("\n");
}

int main()
{
    int arr[2];
    printf("Enter lower limit: ");
    scanf("%d", &arr[0]);
    printf("Enter upper limit: ");
    scanf("%d", &arr[1]);
    pthread_t thread;
    pthread_create(&thread, 0, &prime, (void*)arr);
    pthread_join(thread, NULL);
}

```

output:



```

student@plab-Lenovo-Product: ~/Desktop/180905105/180905105_labs/OS/week4$ gcc 18_3.c -lpthread
student@plab-Lenovo-Product: ~/Desktop/180905105/180905105_labs/OS/week4$ ./a.out
Enter lower limit: 2
Enter upper limit: 7
Prime numbers: 2 3 5 7
student@plab-Lenovo-Product: ~/Desktop/180905105/180905105_labs/OS/week4$

```

Q4)Write a multithread program that performs the sum of even numbers and odd numbers inan input array

```

#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <string.h>

void * even(void *brr)
{
    int *arr = (int*)brr;
    int size = arr[0];
    int sum = 0;
    for (int i = 1; i <= size; i++)
        if (arr[i] % 2 == 0)
            sum += arr[i];
    return (void *) sum;
}

```

```

void * odd(void *brr)
{
    int *arr = (int*)brr;
    int size = arr[0];
    int sum = 0;
    for (int i = 1; i <= size; i++)
        if (arr[i] % 2 != 0)
            sum += arr[i];
    return (void *) sum;
}

int main()
{
    int n, e, o;
    printf("Enter size of array: ");
    scanf("%d", &n);
    int arr[n + 1];
    arr[0] = n;
    printf("Enter elements:\n");
    for (int i = 1; i <= n; i++)
        scanf("%d", &arr[i]);

    pthread_t t1, t2;
    pthread_create(&t1, 0, &even, (void *)arr);
    pthread_create(&t2, 0, &odd, (void *)arr);
    pthread_join(t1, (void*) &e);
    pthread_join(t2, (void*) &o);
    printf("Sum of even numbers = %d\n", (int)e);
    printf("Sum of odd numbers = %d\n", (int)o);
}

```

OUTPUT:

```

l8_4.c: In function 'even':
l8_4.c:14:9: warning: cast to pointer from integer of different size [-Wint-to-p
pointer-cast]
    return (void *) sum;
           ^
l8_4.c: In function 'odd':
l8_4.c:25:9: warning: cast to pointer from integer of different size [-Wint-to-p
pointer-cast]
    return (void *) sum;
           ^
student@lplab-Lenovo-Product:~/Desktop/180905105/180905105_labs/OS/week4$ ./a.out
Enter size of array: 7
Enter elements:
3
5
6
2
1
5
3
Sum of even numbers = 8
Sum of odd numbers = 17
student@lplab-Lenovo-Product:~/Desktop/180905105/180905105_labs/OS/week4$

```