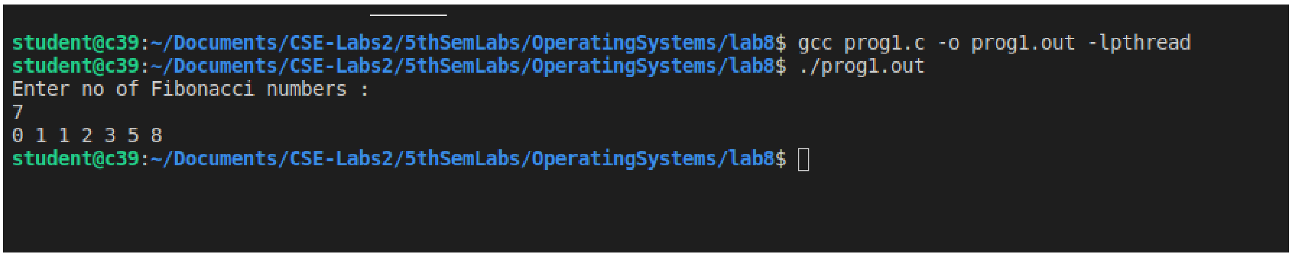
Lab 8 OS

Sahil Saini Salaria
Reg No. 180905048
Roll No. 11C
Batch 5

Q1

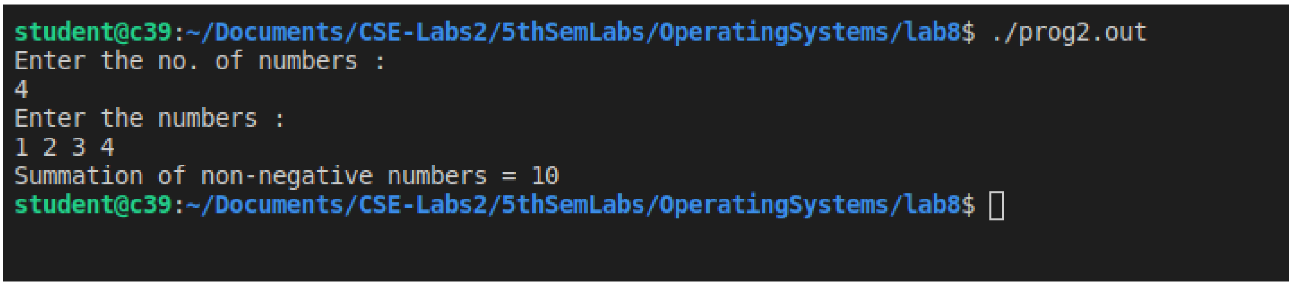
```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>

void* generate_fibonacci(void* param) {
  int* arr = (int*)param;
  int n = arr[0];
  arr[1] = 0;
  arr[2] = 1;
  for(int i = 3;i <= n;i++) {
    arr[i] = arr[i-1] + arr[i-2];
  }
  return NULL;
}

int main(int argc, char const *argv[])
{
  int n;
  printf("Enter no of Fibonacci numbers : \n");
  scanf("%d",&n);

  int* arr = (int*)malloc((n+1)*sizeof(int));
  arr[0] = n;

  pthread_t thread;
  pthread_create(&thread,0,&generate_fibonacci,(void*)arr);
  pthread_join(thread,0);

  for(int i = 1;i <= n;i++)
    printf("%d ",arr[i]);
  printf("\n");
  return 0;
}
```



```
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ gcc prog1.c -o prog1.out -lpthread
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ ./prog1.out
Enter no of Fibonacci numbers :
7
0 1 1 2 3 5 8
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ 
```

Q2

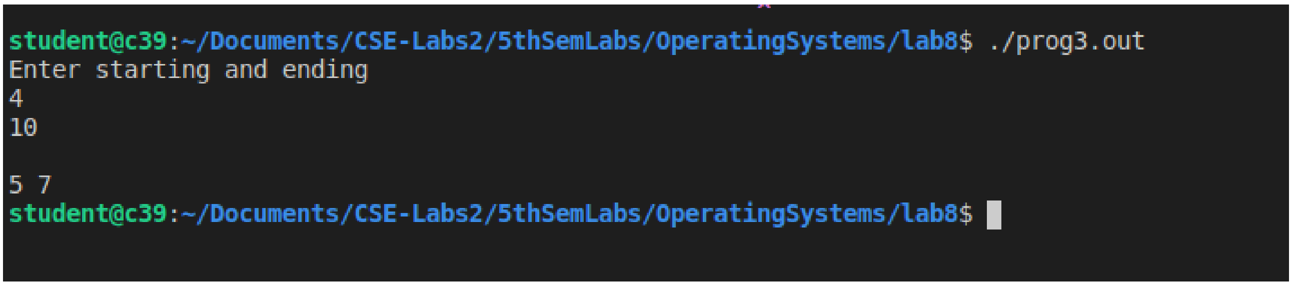
```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>

void* summation(void* param) {
  int* arr = (int*)param;
  int sum = 0;
  int n = arr[0];
  for(int  i = 1;i <= n;i++) {
    if(arr[i] > 0)
      sum += arr[i];
  }
  return (void*)sum;
}

int main(int argc, char const *argv[])
{
  int n;
  printf("Enter the no. of numbers : \n");
  scanf("%d",&n);

  int* arr = (int*)malloc((n+1)*sizeof(int));
  arr[0] = n;

  printf("Enter the numbers : \n");
  for(int  i= 1;i <= n;i++) {
    scanf("%d",&arr[i]);
  }

  int answer = 0;
  pthread_t thread;
  pthread_create(&thread,0,&summation,(void*)arr);
  pthread_join(thread,(void**)&answer);

  printf("Summation of non-negative numbers = %d\n",answer);
  return 0;
}
```



```
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ ./prog2.out
Enter the no. of numbers :
4
Enter the numbers :
1 2 3 4
Summation of non-negative numbers = 10
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ 
```

Q3

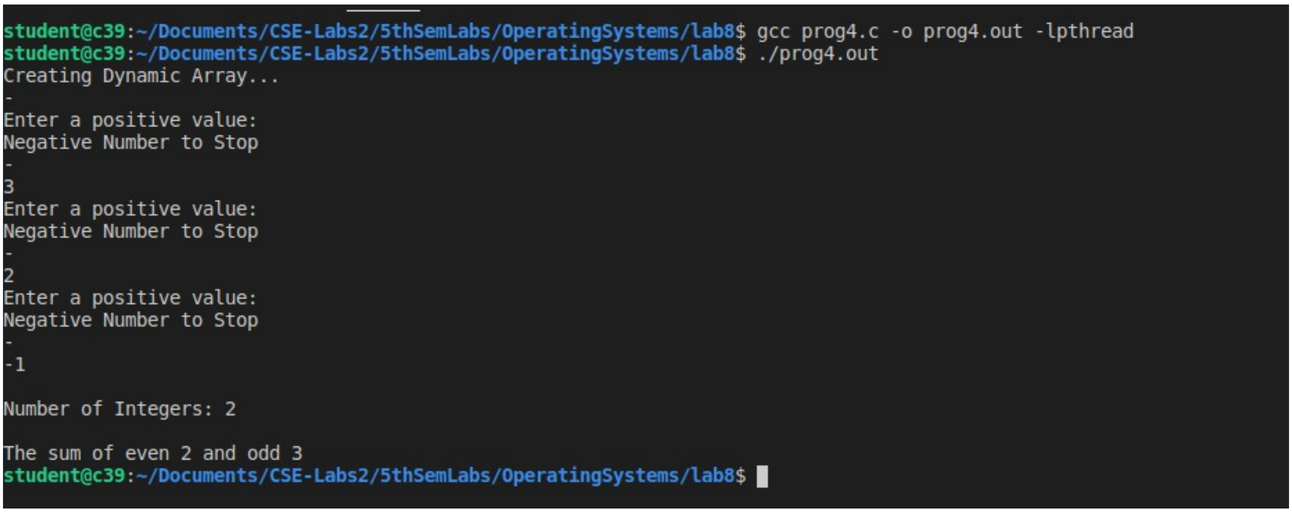
```
#include<stdio.h>
#include<pthread.h>

#define N 30
#define MAX_THREADS 4

int prime_arr[N]={0};

void *printprime(void *ptr)
{
    int  j,flag;
    int i=(int)(long long int)ptr;
    while(i<N)
    {
        // printf("Thread id[%d] checking [%d]\n",pthread_self(),i);
        flag=0;
        for(j=2;j<=i/2;j++)
        {
            if(i%j==0)
            {
                flag=1;
                break;
            }
        }

        if(flag==0 && (i>1))
        {
            prime_arr[i]=1;
        }
        i+=MAX_THREADS;
  }
}

int main()
{
    pthread_t tid[MAX_THREADS]={{0}};
    int count=0;
    printf("Enter starting and ending\n");
    int st,en;
    scanf("%d %d",&st,&en);

    for(count=0;count<MAX_THREADS;count++)
    {
        // printf("\r\n CREATING THREADS %d",count);
        pthread_create(&tid[count],NULL,printprime,(void*)count);
    }
    printf("\n");
    for(count=0;count<MAX_THREADS;count++)
    {
        pthread_join(tid[count],NULL);
    }

    int c=0;

    for(count=st;count<en;count++)
        if(prime_arr[count]==1)
            printf("%d ",count);
   printf("\n");

    return 0;
 }
```



```
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ ./prog3.out
Enter starting and ending
4
10

5 7
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ 
```

Q4

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <errno.h>
#include <ctype.h>
#include <unistd.h>

#define handle_error_en(en, msg) \
        do { errno = en; perror(msg); exit(EXIT_FAILURE); } while (0)

volatile int running_threads = 0;
pthread_t thread[1];
int numOfElements;
struct Results{
  int sum;
}Results,Results2;
void *findsum(void *array_ptr){

  int i;   /*counter*/

  int *elements = (int*)array_ptr;
  for(i = 0; i < numOfElements; i++){
    if(elements[i]%2==0)
    Results.sum += elements[i];
     else
    Results2.sum += elements[i];
  }

  running_threads -= 1;
return NULL;
}
int getArrayInput(int n, int *array_ptr){
    int input;
    int numberOfElements = 0;
      printf("Creating Dynamic Array...\n-\n");
    for(;;){
        printf("Enter a positive value:\nNegative Number to Stop\n-\n");

      if (scanf("%d",&input) != 1){
        printf("\nOops that wasn't an Integer\nlets try filling the array again\nRemember INTEGERS only!\n");
        exit(EXIT_FAILURE);
      }
        if (input >= 0){
            if (numberOfElements == n){
                  n += 1;
                array_ptr = realloc(array_ptr, n * sizeof(int));
             }
            array_ptr[numberOfElements++] = input;
        } else {
           printf("\nNumber of Integers: %d\n", numberOfElements);
           break;
           }
      }
  return numberOfElements;

    }


void createThreads(int *array_ptr){

  int s;

   s = pthread_create(&thread[2], NULL, findsum, (void *)array_ptr);

     if (s != 0){

           handle_error_en(s, "pthread_create");

         }

      running_threads += 1;

}
int main(){
  int n = 1;

  int *array_ptr = malloc(n * sizeof(int));

     numOfElements = getArrayInput(n, array_ptr);

     createThreads(array_ptr);

        while(running_threads>0){

        sleep(1);

      }
    printf("\nThe sum of even %d and odd %d\n",Results.sum,Results2.sum);
  return(0);
}
```



```
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ gcc prog4.c -o prog4.out -lpthread
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ ./prog4.out
Creating Dynamic Array...
-
Enter a positive value:
Negative Number to Stop
-
3
Enter a positive value:
Negative Number to Stop
-
2
Enter a positive value:
Negative Number to Stop
-
-1

Number of Integers: 2

The sum of even 2 and odd 3
student@c39:~/Documents/CSE-Labs2/5thSemLabs/OperatingSystems/lab8$ 
```