**ANOUSHKA GUPTA  
180905424**

**CSE-D,Roll no 51**

**BATCH B9**

**1. Fibonacci series using multithreading**

#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 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);

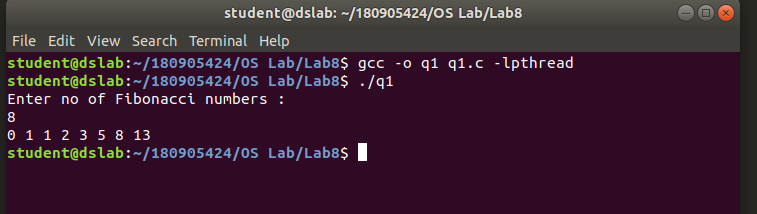
for(int i = 1;i <= n;i++)

printf("%d ",arr[i]);

printf("\n");

return 0;

}



**2. Multi-threaded prgram which finds summation of non negative numbers**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

void\* summation(void\* a)

{

int \* arr=(int\*)a;

int n=arr[0];

int sum=0;

for(int i=1;i<n+1;i++)

sum+=arr[i];

return ((void\*)sum);

}

int main()

{

int n;

printf("Enter the number of numbers:\n");

scanf("%d",&n);

int \*arr= (int\*)malloc(sizeof(int)\*(n+1));

arr[0]=n;

printf("Enter the number:\n");

for(int i=1;i<n+1;i++)

scanf("%d",&arr[i]);

int ans=0;

pthread\_t thread;

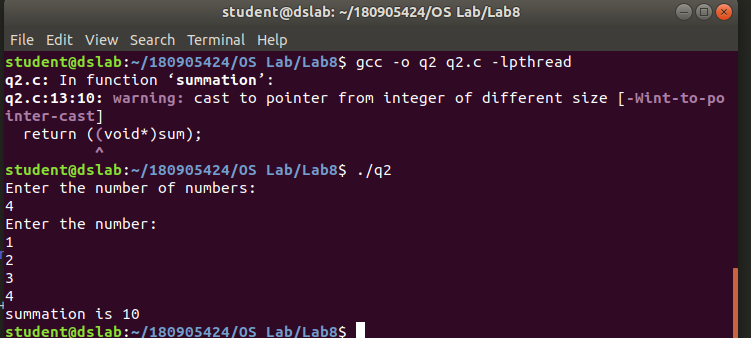
pthread\_create(&thread,NULL,&summation,(void\*)arr);

pthread\_join(thread,(void\*\*)&ans);

printf("summation is %d\n",ans);

return 0;

}



**3. Generate prime numbers**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<math.h>

#define MAX\_SIZE 256

typedef struct{

int m;

int n;

int prime\_count;

int primes[MAX\_SIZE];

}prime;

void\* generate\_primes(void\* param) {

prime\* parameter = (prime\*)param;

//generate primes

for(int i = parameter->m;i <= parameter->n;i++) {

int root = sqrt(i);

int j;

for(j = 2;j <= root;j++) {

if(i % j == 0)

break;

}

if(j == root+1) {

parameter->primes[parameter->prime\_count] = i;

parameter->prime\_count += 1;

}

}

return NULL;

}

int main()

{

prime p;

printf("Enter lower limit :\n");

scanf("%d",&(p.m));

printf("Enter upper limit :\n");

scanf("%d",&(p.n));

p.prime\_count = 0;

pthread\_t thread;

pthread\_create(&thread,0,&generate\_primes,(void\*)&p);

pthread\_join(thread,NULL);

for(int i = 0;i < p.prime\_count;i++) {

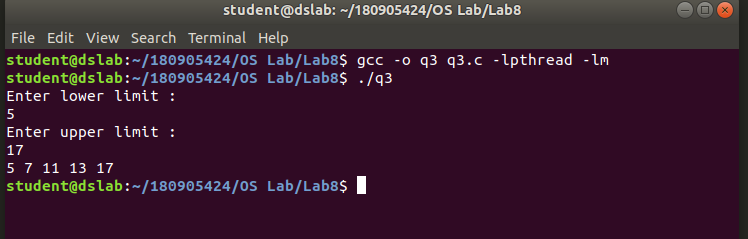
printf("%d ",p.primes[i]);

}

printf("\n");

return 0;

}



**4. Sum of even and odd numbers in array**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

void\* sum\_odd(void\* a)

{

int \* arr=(int\*)a;

int n=arr[0];

int sum=0;

for(int i=1;i<n+1;i++)

{

if(arr[i]%2!=0)

sum+=arr[i];

}

return ((void\*)sum);

}

void\* sum\_even(void\* a)

{

int \* arr=(int\*)a;

int n=arr[0];

int sum=0;

for(int i=1;i<n+1;i++)

{

if(arr[i]%2==0)

sum+=arr[i];

}

return ((void\*)sum);

}

int main()

{

int n;

printf("Enter the number of numbers:\n");

scanf("%d",&n);

int \*arr= (int\*)malloc(sizeof(int)\*(n+1));

arr[0]=n;

printf("Enter the number:\n");

for(int i=1;i<n+1;i++)

scanf("%d",&arr[i]);

int ans1=0,ans2=0;

pthread\_t thread1,thread2;

pthread\_create(&thread1,NULL,&sum\_odd,(void\*)arr);

pthread\_join(thread1,(void\*\*)&ans1);

pthread\_create(&thread2,NULL,&sum\_even,(void\*)arr);

pthread\_join(thread2,(void\*\*)&ans2);

printf("Sum of odd numbers in array is %d\n",ans1);

printf("Sum of even numbers in array is %d\n",ans2);

return 0;

}

