Assignment 2

-Harsh Bansal

-2016UCP1468

1. Write a Program to sum of first n numbers using thread in C.

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <math.h>

void \*sum(void \*inp)

{

long long \*n=(long long \*)inp;long long i,s=0;

long long no=\*n;

for(i=0;i<=no;i++)

s+=i;

printf("%lld\n",s);

}

int main(int argc, char \*\*argv) {

long long n=atoll(argv[1]);

pthread\_t thread;

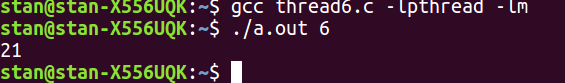
pthread\_create(&thread,NULL,sum,&n);

pthread\_join(thread,NULL);

return 0;

}

Output:



2.Write a Program to find factorial of a given number using thread in C.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <math.h>

void \*sum(void \*inp)

{

long long \*n=(long long \*)inp;long long i,s=1;

long long no=\*n;

for(i=1;i<=no;i++)

s=s\*i;

printf("%lld\n",s);

}

int main(int argc, char \*\*argv) {

long long n=atoll(argv[1]);

pthread\_t thread;

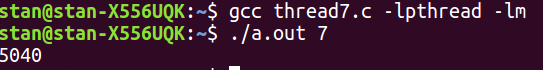
pthread\_create(&thread,NULL,sum,&n);

pthread\_join(thread,NULL);

return 0;

}

Output:



3.Write a Program to find max and min number in an array using threads in C.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <math.h>

struct array

{

int n;

int arr[100];

};

void \*fun(void \*a)

{

struct array \*first=(struct array \*)a;int min=first->arr[0];int max=first->arr[0];int i;

for(i=0;i<first->n;i++)

{

if(min>first->arr[i])

min=first->arr[i];

if(max<first->arr[i])

max=first->arr[i];

}

printf("%d\n",max);

printf("%d\n",min);

}

int main(int argc, char const \*argv[]) {

int n=argc-1;

struct array a;

a.n=n;int i;

for(i=0;i<n;i++)

a.arr[i]=atoi(argv[i+1]);

pthread\_t thread;

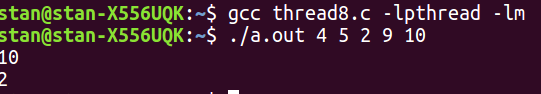
pthread\_create(&thread,NULL,fun,&a);

pthread\_join(thread,NULL);

return 0;

}

Output:



4.Take Student Information(Name , ID, Subject , Marks) from user and find the % of the student.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <math.h>

struct Student

{

const char \*Name;

const char \*ID;

const char \*Subject[100];

int Marks[100];

int n;

};

void \*percentage(void \*a)

{

struct Student \*stud=(struct Student \*)a;double sum=0,perc;int i;

for(i=0;i<stud->n;i++)

{

sum=sum+stud->Marks[i];

}

perc=sum/stud->n;

printf("%.2f\n",perc);

}

int main(int argc, char const \*argv[]) {

int n=(argc-3)/2;

struct Student student;

student.n=n;

student.Name=argv[1];

student.ID=argv[2];int i;

for(i=0;i<n;i++)

{

student.Subject[i]=argv[2\*i+3];

student.Marks[i]=atoi(argv[2\*i+4]);

}

pthread\_t thread;

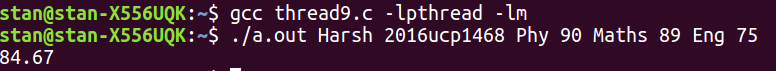
pthread\_create(&thread,NULL,percentage,&student);

pthread\_join(thread,NULL);

return 0;

}

Output:



5.Take Employee Information(Name, ID, Designation, Salary of last 5 years ) from user and find the % hike in salary of the employee using Thread and Mutex lock (use structure).

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <math.h>

#include <unistd.h>

int count=0;

pthread\_mutex\_t lock;

struct Employee

{

const char \*Name;

const char \*ID;

const char \*Designation;

int Salary[5];

};

void \*Hike(void \*a)

{

struct Employee \*stud=(struct Employee \*)a;double hike;long long i;int b;

pthread\_mutex\_lock(&lock);

hike= (stud->Salary[count+1]-stud->Salary[count])\*100/stud->Salary[count];

for(i=0;i<100;i++)

b=1;

count++;

printf(" Hike after %d year : %.2f\n",count,hike);

pthread\_mutex\_unlock(&lock);

}

int main(int argc, char const \*argv[]) {

pthread\_mutex\_init(&lock,NULL);

struct Employee employee;

employee.Name=argv[1];

employee.ID=argv[2];

employee.Designation=argv[2];

int i;

for(i=0;i<5;i++)

{

employee.Salary[i]=atoi(argv[i+4]);

}

pthread\_t thread1,thread2,thread3,thread4;

pthread\_create(&thread1,NULL,Hike,&employee);

pthread\_create(&thread2,NULL,Hike,&employee);

pthread\_create(&thread3,NULL,Hike,&employee);

pthread\_create(&thread4,NULL,Hike,&employee);

pthread\_join(thread1,NULL);

pthread\_join(thread2,NULL);

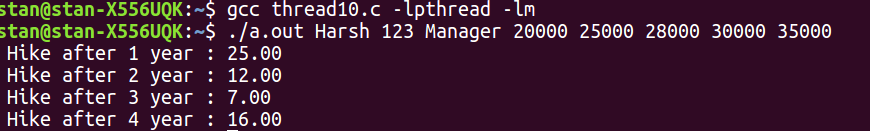
pthread\_join(thread3,NULL);

pthread\_join(thread4,NULL);

return 0;

}

Output:



6. Write a Program to calculate Prime numbers series and Fibonacci series for a given number

using Thread and Mutex Lock.

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

int fib[10000];

int prime[10000];

int idx=0;

pthread\_mutex\_t mutex1=PTHREAD\_MUTEX\_INITIALIZER;

pthread\_mutex\_t mutex2=PTHREAD\_MUTEX\_INITIALIZER;

void\* fiboni(void \*arg)

{

int \* num = (int \*)arg;

pthread\_mutex\_lock(&mutex1);

fib[0]=0;

fib[1]=1;

int i;

for(i=2;i<\*num;i++)

{

fib[i]=fib[i-1]+fib[i-2];

}

pthread\_mutex\_unlock(&mutex1);

}

void\* primecheck(void \*arg)

{

int \* num = (int \*)arg;

pthread\_mutex\_lock(&mutex2);

int i,j;

for(i=1; i<=\*num; i++)

{

int factor=0;

for(j=1; j<=\*num; j++)

{

if(i%j==0)

factor++;

}

if(factor==2)

{

prime[idx]=i;

idx++;

}

}

pthread\_mutex\_unlock(&mutex2);

}

int main()

{

int i;

pthread\_t thread1,thread2;

int n;

printf("enter the number\n");

scanf("%d",&n);

pthread\_create(&thread1,NULL,fiboni,(void \*)&n);

pthread\_create(&thread2,NULL,primecheck,(void \*)&n);

pthread\_join(thread1,NULL);

pthread\_join(thread2,NULL);

printf("Fibonacci series for %d is:: \n",n);

for(i=0;i<n;i++)

{

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

}

printf("\n");

printf("Prime Number Series for %d is:: \n",n);

for(i=0;i<idx;i++)

{

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

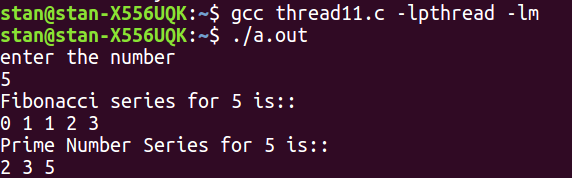
}

printf("\n");

return 0;

}

Output:



7.Write a Program to computes the total of the values of the matrix using thread in C.

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<math.h>

int arr[10][10];

int l,m,sum=0,total;

int counter=-1;

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

void\* func(void\* arg)

{

pthread\_mutex\_lock(&lock);

int i;

counter++;

total=0;

for(i=0;i<m;i++)

{

total+=arr[counter][i];

}

sum+=total;

//printf("%d\n",sum);

pthread\_mutex\_unlock(&lock);

return NULL;

}

int main(int argc,char\* argv[])

{

int i,j;

printf("enter row and column\n");

scanf("%d %d",&l,&m);

printf("fill array\n");

for(i=0;i<l;i++)

for(j=0;j<m;j++)

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

pthread\_t thread[l];

for(i=0;i<l;i++)

{

pthread\_create(&thread[i],NULL,&func,NULL);

}

for(i=0;i<l;i++)

pthread\_join(thread[i],NULL);

printf("%d\n",sum);

return 0;

}

Output:

