**CPP LAB**

**ASSIGNMENT-2**

**NAME-**ABHISHEK TIBREWAL

**ID-**2016UCP1103

**BATCH-**A (1, 2)

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

**CODE:**

# include <stdio.h>

# include <pthread.h>

int Total=0;

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

void \*sum(void \*x)

{

int \*i,c;

i=(int\*)x;

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

{

pthread\_mutex\_lock(&flag);

Total=Total+ c;

pthread\_mutex\_unlock(&flag);

}

pthread\_exit(NULL);

}

main()

{

int n,c;

pthread\_t t;

printf("Enter Number:");

scanf("%d",&n);

pthread\_create(&t,NULL,sum,(void \*)&n);

for(c=1;c<=n;c=c+2)

{

pthread\_mutex\_lock(&flag);

Total=Total+c;

pthread\_mutex\_unlock(&flag);

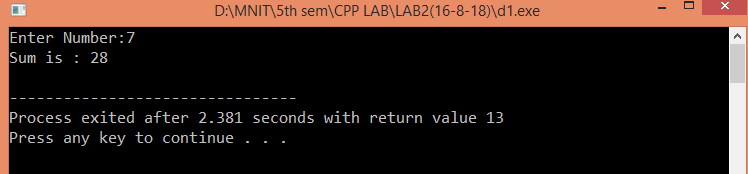
}

pthread\_join(t,NULL);

printf("Sum is : %d \n",Total);

}

**OUTPUT:**

****

1. **Write a Program find factorial of a given number using thread in C.**

**CODE:**

# include <stdio.h>

# include <pthread.h>

int Total=1;

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

void \*fact(void \*x)

{

int \*i,c;

i=(int\*)x;

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

{

pthread\_mutex\_lock(&flag);

Total=Total\*c;

pthread\_mutex\_unlock(&flag);

}

pthread\_exit(NULL);

}

main()

{

int n,c;

pthread\_t t;

printf("Enter Number:");

scanf("%d",&n);

pthread\_create(&t,NULL,fact,(void \*)&n);

for(c=1;c<=n;c=c+2)

{

pthread\_mutex\_lock(&flag);

Total=Total\*c;

pthread\_mutex\_unlock(&flag);

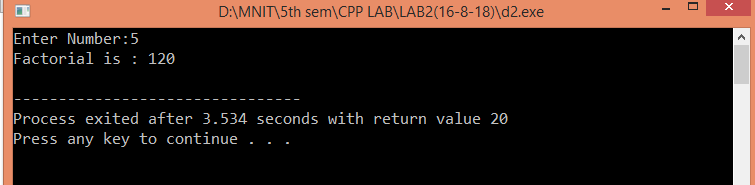
}

pthread\_join(t,NULL);

printf("Factorial is : %d \n",Total);

}

**OUTPUT:**

****

1. **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<limits.h>

int \*inp;

int n,c;

int max=INT\_MIN;

int min=INT\_MAX;

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

void \*sum(void \*x)

{

for(c=0;c<n;c=c+2)

{

pthread\_mutex\_lock(&flag);

if(max<inp[c])

{

max=inp[c];

}

if(min>inp[c])

{

min=inp[c];

}

pthread\_mutex\_unlock(&flag);

}

pthread\_exit(NULL);

}

main()

{

int i;

pthread\_t t;

printf("Enter Number of elements in array:");

scanf("%d",&n);

inp=(int \*)malloc(sizeof(int)\*n);

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

{

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

}

pthread\_create(&t,NULL,sum,(void \*)inp);

for(c=1;c<n;c=c+2)

{

pthread\_mutex\_lock(&flag);

if(max<inp[c])

{

max=inp[c];

}

if(min>inp[c])

{

min=inp[c];

}

pthread\_mutex\_unlock(&flag);

}

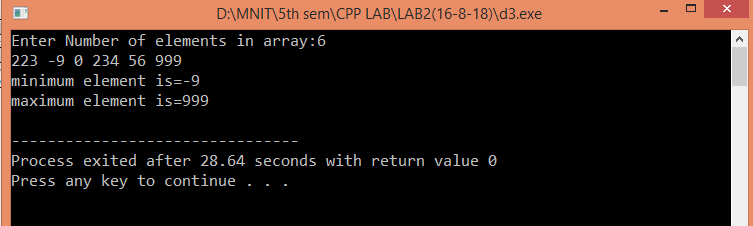
pthread\_join(t,NULL);

printf("minimum element is=%d\nmaximum element is=%d\n",min,max);

return 0;

}

**OUTPUT:**

****

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

**CODE:**

# include <stdio.h>

# include <pthread.h>

#include<stdlib.h>

int Total=0;

int \*s;

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

void \*sum(void \*x)

{

int \*i,c;

i=(int\*)x;

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

{

pthread\_mutex\_lock(&flag);

Total=Total+ s[c-1];

pthread\_mutex\_unlock(&flag);

}

pthread\_exit(NULL);

}

main()

{

char name[20];

int id,ns,i,c;

pthread\_t t;

printf("Enter name,id,total number of subjects:\n");

scanf("%s%d%d",name,&id,&ns);

s=(int \*)malloc(sizeof(int)\*ns);

printf("enter marks in each subject:\n");

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

{

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

}

pthread\_create(&t,NULL,sum,(void \*)&ns);

for(c=1;c<=ns;c=c+2)

{

pthread\_mutex\_lock(&flag);

Total=Total+s[c-1];

pthread\_mutex\_unlock(&flag);

}

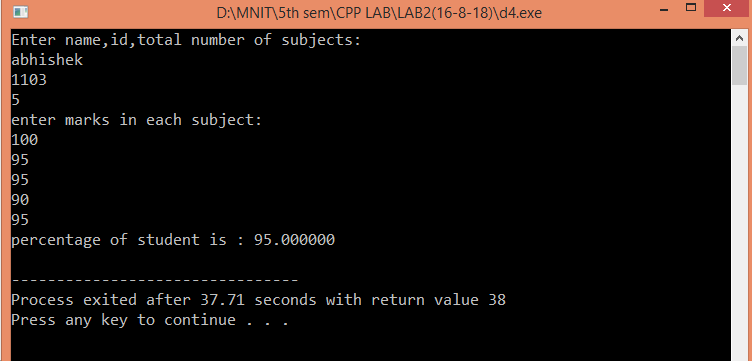
pthread\_join(t,NULL);

float p=(Total\*100)/(100\*ns);

printf("percentage of student is : %f \n",p);

}

**OUTPUT:**

****

1. **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).**

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

struct EMPLOYEE

{

char Ename[50];

char Eid[11];

char Edesignation[30];

int sal[5];

};

struct EMPLOYEE \*p;

pthread\_mutex\_t t1 = PTHREAD\_MUTEX\_INITIALIZER;

void\* fun(void\* x)

{

pthread\_mutex\_lock(&t1);

int i= (int)x;

float hike = (float)(p->sal[i]-p->sal[i-1])/(p->sal[i-1]);

hike = hike\*100;

pthread\_mutex\_unlock(&t1);

return (void\* )(int)(hike);

}

int main()

{

p=(struct EMPLOYEE \*)malloc(sizeof(struct EMPLOYEE));

printf("Enter name of employee : ");

scanf("%s",p->Ename);

printf("\nEnter ID of employee : ");

scanf("%s",p->Eid);

printf("\nEnter designation of employee : ");

scanf("%s",p->Edesignation);

printf("\nEnter salary of the last five years\n");

int i;

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

{

scanf("%d",&p->sal[i]);

}

pthread\_t th[4];

for(i=1;i<5;i++)

{

pthread\_create(&th[i],NULL,fun,(void\*)i);

}

void\* ANS;

for(i=1;i<5;i++)

{

pthread\_join(th[i],&ANS);

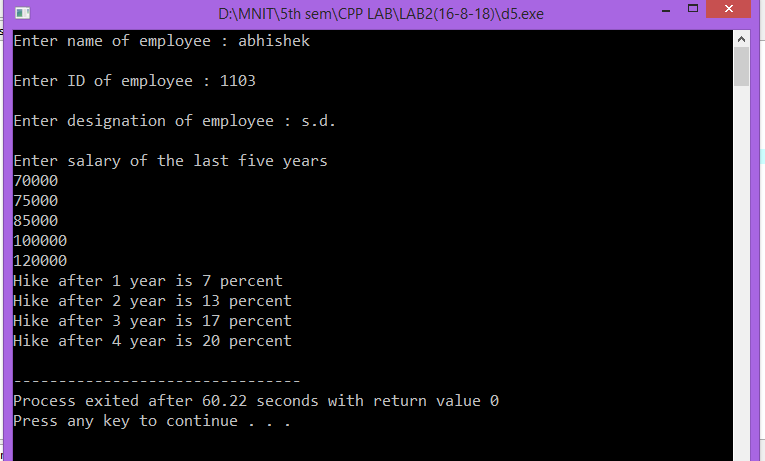
printf("Hike after %d year is %d percent\n",i,(int)ANS);

}

return 0;

}

**OUTPUT:**

****

1. **Write a Program to calculate Prime numbers series and Fibonacci series for a given number using Thread and Mutex Lock.**

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <string.h>

int p[10000];

int f[10000];

int n,c;

pthread\_mutex\_t t1 = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_mutex\_t t2 = PTHREAD\_MUTEX\_INITIALIZER;

void\* funprime()

{

int i,j,fact;

pthread\_mutex\_lock(&t1);

c=0;

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

{

fact=0;

for(j=1; j<=n; j++)

{

if(i%j==0)

fact++;

}

if(fact==2)

{

p[c]=i;

c++;

}

}

pthread\_mutex\_unlock(&t1);

}

void\* funfib()

{

pthread\_mutex\_lock(&t2);

f[0]=0;

f[1]=1;

int i;

for(i=2;i<n;i++)

{

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

}

pthread\_mutex\_unlock(&t2);

}

int main()

{

pthread\_t th1,th2;

printf("enter value of n\n");

scanf("%d",&n);

pthread\_create(&th1,NULL,funprime,NULL);

pthread\_create(&th2,NULL,funfib,NULL);

pthread\_join(th1,NULL);

pthread\_join(th2,NULL);

int i;

printf("Fibonacci series\n");

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

{

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

}

printf("\nPrime Number Series\n");

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

{

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

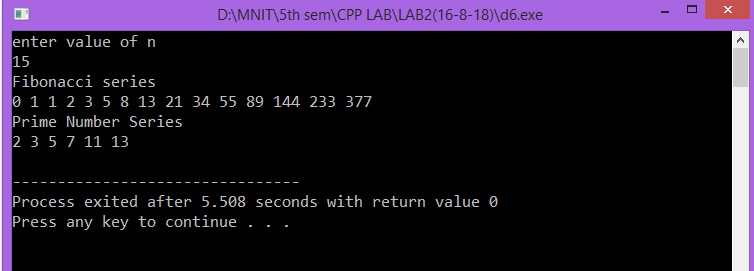
}

printf("\n");

return 0;

}

**OUTPUT:**

****

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

**CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<math.h>

int inp[10][10];

int r,c,s=0,f;

int co=-1;

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

void \* fun(void \*x) //function to be run when thread is created.

{

pthread\_mutex\_lock(&flag);

int i;

co++;

f=0;

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

{

f+=inp[co][i];

}

sleep(3);

s=s+f;

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

pthread\_mutex\_unlock(&flag);

return NULL;

}

int main()

{

int x,y;

printf("enter no. of rows and columns respectively\n");

scanf("%d%d",&r,&c);

printf("enter the array elements\n");

for(x=0;x<r;x++)

{

for(y=0;y<c;y++)

{

scanf("%d",&inp[x][y]);

}

}

pthread\_t t[r];

for(x=0;x<r;x++)

{

pthread\_create(&t[x],NULL,&fun,NULL);

}

for(x=0;x<r;x++)

{

pthread\_join(t[x],NULL);

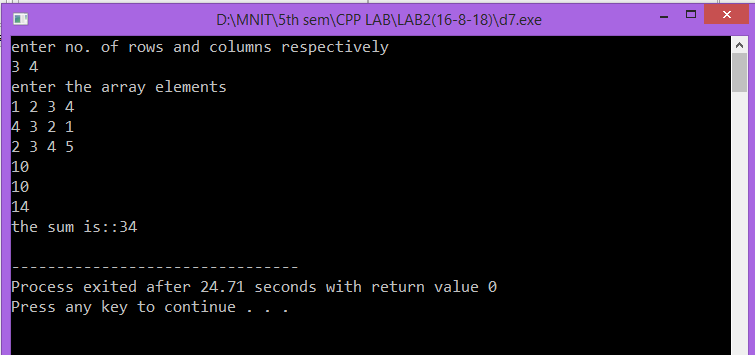
}

printf("the sum is::%d\n",s);

return 0;

}

**OUTPUT:**

****