**CPP Lab**

**Assignment-2**

**Name-Ashish Goyal**

**Id-2016ucp1100**

**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>

#include<stdlib.h>

int ans=0;

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

void \*func(void \*arg)

{

int \*n,i;

n=(int\*)arg;

for(i=2;i<=\*n;i=i+2)//claculate even sum

{

pthread\_mutex\_lock(&lock);

ans=ans+i;

pthread\_mutex\_unlock(&lock);

}

pthread\_exit(NULL);

}

main()

{

int n,i;

pthread\_t thread;

printf("Enter Natural Number\n");

scanf("%d",&n);

pthread\_create(&thread,NULL,func,(void \*)&n);

for(i=1;i<=n;i=i+2)//calculate odd sum

{

pthread\_mutex\_lock(&lock);

ans=ans+i;

pthread\_mutex\_unlock(&lock);

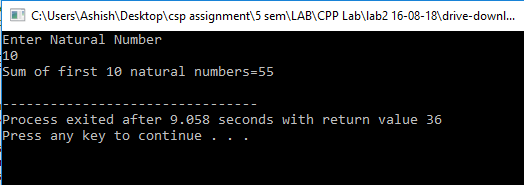
}

pthread\_join(thread,NULL);

printf("Sum of first %d natural numbers=%d \n",n,ans);

}

**OUTPUT-**

****

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

**CODE-**

#include<stdio.h>

#include<pthread.h>

#include<stdlib.h>

int ans=1;

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

void \*func(void \*arg)

{

int \*n,i;

n=(int\*)arg;

for(i=2;i<=\*n;i=i+2)//claculate even

{

pthread\_mutex\_lock(&lock);

ans=ans\*i;

pthread\_mutex\_unlock(&lock);

}

pthread\_exit(NULL);

}

main()

{

int n,i;

pthread\_t thread;

printf("Enter Natural Number\n");

scanf("%d",&n);

pthread\_create(&thread,NULL,func,(void \*)&n);

for(i=1;i<=n;i=i+2)//calculate odd

{

pthread\_mutex\_lock(&lock);

ans=ans\*i;

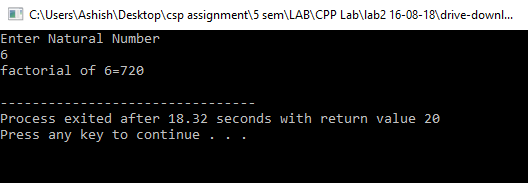
pthread\_mutex\_unlock(&lock);

}

pthread\_join(thread,NULL);

printf("factorial of %d=%d \n",n,ans);

}

**OUTPUT-**

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

**CODE-**

#include<stdio.h>

#include<pthread.h>

#include<stdlib.h>

#include<limits.h>

int max=INT\_MIN,min=INT\_MAX;

int \*arr;

int n;

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

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

void \*func(void \*arg)

{

int i;

for(i=0;i<n;i=i+2)//claculate even sum

{

pthread\_mutex\_lock(&lock);

if(min>arr[i])

min=arr[i];

if(max<arr[i])

max=arr[i];

pthread\_mutex\_unlock(&lock);

}

return (NULL);

}

main()

{

int i;

pthread\_t thread;

printf("Enter Number of element\n");

scanf("%d",&n);

printf("fill array\n");

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

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

{

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

}

pthread\_create(&thread,NULL,func,(void \*)&n);

for(i=1;i<n;i=i+2)//calculate odd sum

{

pthread\_mutex\_lock(&lock);

if(min>arr[i])

min=arr[i];

if(max<arr[i])

max=arr[i];

pthread\_mutex\_unlock(&lock);

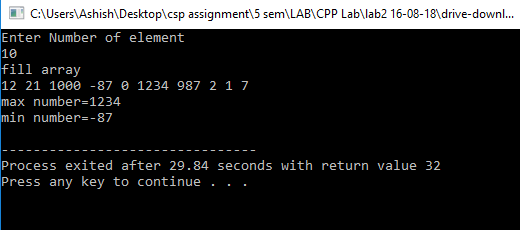
}

pthread\_join(thread,NULL);

printf("max number=%d\nmin number=%d \n",max,min);

}

**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>

#include<limits.h>

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

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

int \*ans;

int sub;

float sum=0.0;

void \*func(void \*arg)

{

int i;

for(i=0;i<sub;i=i+2)//claculate even sum

{

pthread\_mutex\_lock(&lock);

sum=sum+(float)ans[i];

printf("in thread\n");

pthread\_mutex\_unlock(&lock);

}

pthread\_exit(NULL);

}

main()

{

int n,i,id;

char str[20];

pthread\_t thread;

printf("Enter Name,ID,total subjects\n");

scanf("%s%d%d",str,&id,&sub);

printf("enter marks\n");

ans=(int \*)malloc(sizeof(int)\*sub);

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

{

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

}

pthread\_create(&thread,NULL,func,NULL);

for(i=1;i<sub;i=i+2)//calculate odd sum

{

pthread\_mutex\_lock(&lock);

printf("in main\n");

sum=sum+(float)ans[i];

pthread\_mutex\_unlock(&lock);

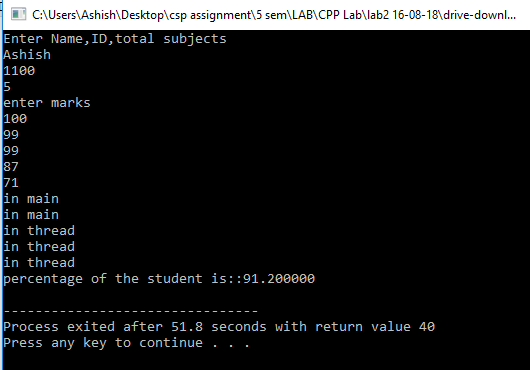
}

pthread\_join(thread,NULL);

printf("percentage of the student is::%f\n",(sum\*1.0)/sub);

}

**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>

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

struct employee

{

char name[20];

char id[20];

char designation[20];

int sal[5];

};

struct employee\* emp;

void\* func(void\* a)

{

pthread\_mutex\_lock(&mutex1);

int temp= (int)a;

float h;

h = ((float)(emp->sal[temp] - emp->sal[temp-1])/(emp->sal[temp-1]))\*100;

pthread\_mutex\_unlock(&mutex1);

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

}

int main()

{

int i;

void\* ans;

emp = (struct employee\*)malloc(sizeof(struct employee));

printf("Name-");

scanf("%s",emp->name);

printf("\nID-");

scanf("%s",emp->id);

printf("\nDesignation-");

scanf("%s",emp->designation);

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

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

{

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

}

pthread\_t thread[4];

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

{

pthread\_create(&thread[i-1],NULL,func,(void\*)i);

}

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

{

pthread\_join(thread[i-1],&ans);

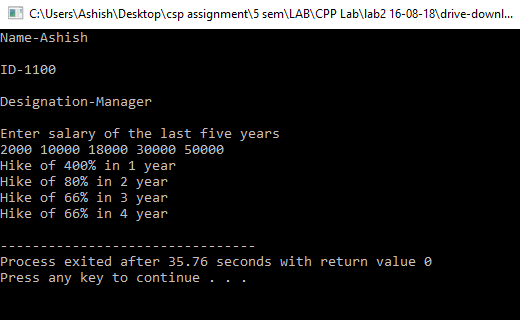
printf("Hike of %d%% in %d year\n",(int)ans,i);

}

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>

int fib[10000];

int prime[10000];

int num,ind=0;

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

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

void\* func(void \*a)

{

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\* func1(void \*a)

{

pthread\_mutex\_lock(&mutex2);

int i,j;

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

{

int fact=0;

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

{

if(i%j==0)

fact++;

}

if(fact==2)

{

prime[ind]=i;

ind++;

}

}

pthread\_mutex\_unlock(&mutex2);

}

int main()

{

int i;

pthread\_t thread1,thread2;

printf("enter the number\n");

scanf("%d",&num);

pthread\_create(&thread1,NULL,func,NULL);

pthread\_create(&thread2,NULL,func1,NULL);

pthread\_join(thread1,NULL);

pthread\_join(thread2,NULL);

printf("Fibonacci series for the given number is:: ");

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

{

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

}

printf("\n");

printf("Prime Number Series for the given number is:: ");

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

{

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

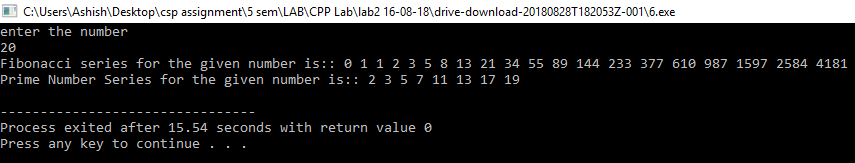
}

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 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];

}

sleep(1);

sum+=total;

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

// sleep(1);

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);

}

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

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

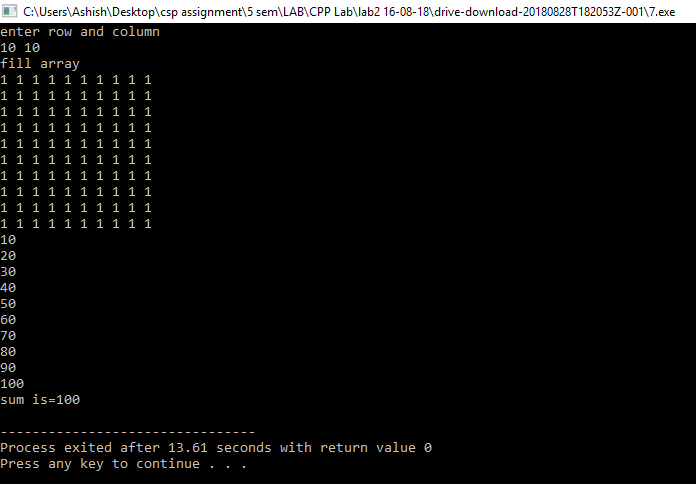
pthread\_join(thread[i],NULL);

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

return 0;

}

**OUTPUT-**

****