**ASSIGNMENT NO.**

NAME: SHRUTI DILIP BHUJANGE

CLASS: BE COMP-1 ROLL NO.: 402006

PROGRAM:

#include<stdio.h>

#include<omp.h>

void InsertionSort(int a[],int n);

int main()

{

int a[20],n,num,i,j;

double start,finish;

printf("\nEnter the number of elements in array:");

scanf("%d", &n);

printf("\nEnter elements of array:\n");

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

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

start=omp\_get\_wtime();

InsertionSort(a,n);

printf("\nSorted array is:\n");

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

printf("%d\t",a[i]);

printf("\n\nEnter the number to be searched:");

scanf("%d", &num);

printf("\n");

int p=BinarySearch(a,0,n-1,num);

finish=omp\_get\_wtime();

if(p==-1)

printf("\nRequired number is not found....!\n");

else

printf("\nRequired number is found at index %d.\n", p);

printf("Time required is %f seconds.\n",finish-start);

return 0;

}

void InsertionSort(int a[],int n)

{

int i,j,temp;

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

{

temp=a[i];

for(j=i-1;j>=0 && a[j]>temp;j--)

{

a[j+1]=a[j];

}

a[j+1]=temp;

}

}

int BinarySearch(int a[],int bottom,int top,int num)

{

int mid,tid,index;

if(bottom<=top)

{

mid = (top + bottom)/2;

#pragma omp parallel sections

{

#pragma omp section

{

tid = omp\_get\_thread\_num();

printf("Thread %d is checking if required number is equal to mid (%d).\n",tid,a[mid]);

if(a[mid]==num)

index= mid;

}

#pragma omp section

{

tid = omp\_get\_thread\_num();

printf("Thread %d is checking if required number is less than mid (%d).\n",tid,a[mid]);

if(num < a[mid])

index= BinarySearch(a,bottom,mid-1,num);

}

#pragma omp section

{

tid = omp\_get\_thread\_num();

printf("Thread %d is checking if required number is greater than mid (%d).\n",tid,a[mid]);

if(num > a[mid])

index= BinarySearch(a,mid+1,top,num);

}

}

}

else

index=-1;

return index;

}