Name- Prerana Rajesh Gajare

Class-SEIT

RollNo-SI41

PROBLEM STATEMENT:-

a) Design a roll call list, arrange the list of students according to roll numbers in

ascending order (Use Bubble Sort)

b) Arrange a list of students alphabetically. (Use Insertion sort)

c) Arrange a list of students to find out the first ten toppers from a class. (Use Quicksort)

d) Search students according to SGPA. If more than one student having the same SGPA,

then print a list of all students having the same SGPA.

e) Search a particular student according to name using binary search without recursion

Source code:-

**#include <iostream>**

**using namespace std;**

**struct Class //Structure class and its variables**

**{**

**int rollno ;**

**string name;**

**float sgpa;**

**};**

**//Input the record from user in the form of array**

**void getdata(struct Class B[20],int m)**

**{**

**int i;**

**cout<<"Name"<<"Rollno"<<"Sgpa";**

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

**{**

**cin>>B[i].name>>B[i].rollno>>B[i].sgpa;**

**}**

**}**

**// (a) i Display the record from user in the form of array.**

**void display(struct Class B[20],int m)**

**{**

**int j;**

**cout<<"\n"<< "Name"<<"\t"<<"Rollno"<<"\t"<<"Sgpa"<<" \t ";**

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

**{**

**cout<<"\n"<<B[j].name<<"\t"<<B[j].rollno<<"\t"<<B[j].sgpa<<"\t";**

**}**

**}**

**// (a) ii Arranging the records of student according to the rollno using Bubble sort.**

**void sort(struct Class B[20],int m)**

**{**

**int i,j;**

**for (i=0;i<m-1;i++)**

**{**

**for (j=0; j<m-i-1; j++)**

**{**

**if(B[j].rollno>B[j+1].rollno)**

**{**

**struct Class hold=B[j];**

**B[j]=B[j+1];**

**B[j+1]=hold;**

**}**

**}**

**}**

**cout<<"Sorted database is(rollno wise): "<<endl;**

**cout<<"Rollno"<<"\t"<<"Name"<<"\t"<<"Sgpa"<<" \t ";**

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

**{**

**cout<<"\n"<<B[i].rollno<<"\t"<<B[i].name<<"\t"<<B[i].sgpa<<"\t";**

**}**

**}**

**// (b) Arranging the records of student Alphabetically using Insertion sort.**

**void insort(struct Class B[20],int m)**

**{**

**int i,j;**

**for(i=1;i<m;i++)**

**{**

**struct Class temp = B[i];**

**j=i-1;**

**while(j>=0 && temp.name<B[j].name)**

**{**

**B[j+1]=B[j];**

**j--;**

**}**

**B[j+1]=temp;**

**}**

**}**

**// (c) To find out toppers from the given record using quick sort.**

**void quicksort(struct Class B[20],int l,int m)**

**{**

**int b,s;**

**int p=l;// Pivot index**

**struct Class temp;**

**if(l<m)**

**{**

**int b=l+1;// Too big index**

**int s=m;// Too small index**

**while(s>b)**

**{**

**while(B[b].sgpa<=B[p].sgpa )//Comparision between data of too big index and pivot index.**

**{**

**++b;**

**}**

**while( B[s].sgpa > B[p].sgpa )//Comparision between data of small big index and pivot index.**

**{**

**--s;**

**}**

**if(b<s)**

**{**

**temp=B[b];// Swaping between data of too big index and too small index.**

**B[b]=B[s];**

**B[s]=temp;**

**}**

**}**

**temp=B[s];// Swaping between data of too small index and pivot index.**

**B[s]=B[p];**

**B[p]=temp;**

**quicksort(B,p,s-1);// First half sorting cycle.**

**quicksort(B,s+1,m);// Second half sorting cycle.**

**}**

**}**

**// (d) Searching records of students through linear search using sgpa.**

**void search(struct Class B[20],int m)**

**{**

**int n,item,i;**

**string again="yes";**

**while(again=="yes" || again=="Yes")//While loop for repeating the search.**

**{**

**cout<<"\nEnter the student's sgpa to search :";**

**cin>>item;**

**int flag=1;//Initializing flag at one.**

**cout<<"\nThe students record having sgpa"<<item<<"is:"<<"\n";**

**cout<<"Name"<<"\t"<<"Rollno"<<"\t"<<"Sgpa"<<" \t ";**

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

**{**

**if(B[i].sgpa==item)//Comparing SGPA Value and data value accepted.**

**{**

**cout<<"\n"<<B[i].name<<"\t"<<B[i].rollno<<"\t"<<B[i].sgpa<<"\t";**

**flag=0;//Once operation is performed set flag to 0.**

**}**

**}**

**if(flag==1)**

**{**

**cout<<"\nSuch student record does not exist";**

**}**

**cout<<"\nDo you want to continue? (yes/no):";**

**cin>>again;**

**}**

**cout<<"The end";**

**}**

**// (e) Searching records of students through binary search using name without recursion.**

**void binsearch(struct Class B[20],int m)**

**{**

**int k=-1,low=0,high=m-1,mid1;//Initializing values**

**string t;**

**cout<<"Enter the name to be searched:";**

**cin>>t;**

**insort(B,m);//Calling insort function.**

**//First half searching cycle.**

**while(low<=high)**

**{**

**mid1=(low+high)/2;//Finding mid.**

**if(B[mid1].name < t)**

**{**

**low=mid1+1;**

**}**

**else if(B[mid1].name==t)**

**{**

**k=mid1;**

**high=mid1-1;**

**}**

**else**

**{**

**high=mid1-1;**

**}**

**}**

**// Second half searching cycle.**

**int l=-1,low1=0,high1=m-1,mid2;//Initializing values.**

**while(low1<=high1)**

**{**

**int mid2=(low1+high1)/2;//Finding mid.**

**if(B[mid2].name<t)**

**{**

**low1=mid2+1;**

**high1=mid2-1;**

**}**

**else if(B[mid2].name==t)**

**{**

**l=mid2;**

**low1=mid2+1;**

**}**

**else**

**{**

**high1=mid2-1;**

**}**

**}**

**if(k!=-1 && l!=-1)**

**{**

**cout<<"Name"<<"\t"<<"Rollno"<<"\t"<<"Sgpa"<<" \t ";**

**for(int i=k;i<=l;i++)**

**{**

**cout<<"\n"<<B[i].name<<"\t"<<B[i].rollno<<"\t"<<B[i].sgpa<<"\t";**

**}**

**}**

**else**

**{**

**cout<<"Such record does not exist in database.";**

**}**

**}**

**// To show first 10 toppers**

**void quickdisplay(struct Class B[20],int m)**

**{**

**int i;**

**int r=10;**

**cout<<"\n"<<"Rollno"<<"\t"<<"Name"<<"\t"<<"Sgpa"<<" \t ";**

**for(i=m-1; i>m-r-1; i--)**

**{**

**cout<<"\n"<<B[i].rollno<<"\t"<<B[i].name<<"\t"<<B[i].sgpa<<"\t";**

**}**

**}**

**// To show user defined number of toppers.**

**void qdisplay(struct Class B[20],int m)**

**{**

**int i;**

**int r;**

**int lm=16;**

**cout<<"Enter the number of toppers you want:";**

**cin>>r;**

**if(r<lm)**

**{**

**cout<<"\n"<<"Rollno"<<"\t"<<"Name"<<"\t"<<"Sgpa"<<" \t ";**

**for(i=m-1; i>m-r-1; i--)**

**{**

**cout<<"\n"<<B[i].rollno<<"\t"<<B[i].name<<"\t"<<B[i].sgpa<<"\t";**

**}**

**}**

**else**

**{**

**cout<<"\nRecheck the value entered";**

**}**

**}**

**int main()**

**{**

**int f,m;**

**struct Class B[20];// Array of Object of structure class**

**struct Class hold;**

**struct Class temp;**

**//Input the number of records to be present in database.**

**cout<<"Enter the number of records to be present:";**

**cin>>m;**

**do{**

**cout<<"\nEnter the function to perform\n1)Input data\n2)Display data\n3)Bubble sort(RollNo. wise)\n4)Insertion sort(Alphabetical wise)\n5)Quick sort(Toppers)\n6)Linear search(SGPA based)\n7)Binary search(Name based)\n8)exit\n(1,2,3,4,5,6,7,8):";**

**cin>>f;**

**switch(f)**

**{**

**case 1:**

**getdata(B,m);//Calling getdata function**

**break;**

**case 2:**

**display(B,m);//Calling display function**

**break;**

**case 3:**

**sort(B,m);//Calling sort function to perform bubble sort.**

**break;**

**case 4:**

**insort(B,m);//Calling insort function to perform insertion sort.**

**cout<<"Sorted database is (alphabetical wise): "<<endl;**

**cout<<"Rollno"<<"\t"<<"Name"<<"\t"<<"Sgpa"<<" \t ";**

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

**{**

**cout<<"\n"<<B[i].rollno<<"\t"<<B[i].name<<"\t"<<B[i].sgpa<<"\t";**

**}**

**break;**

**case 5:**

**int x;**

**do**

**{**

**cout<<"\nChoose the form of output you want\n10)First 10 toppers\n20)User defined number of toppers\n30)exit\n(10,20,30):";**

**cin>>x;**

**switch(x)**

**{**

**case 10:**

**quicksort(B,0,m-1);//Calling quicksort function**

**quickdisplay(B,m);// to display first 10 toppers from the record.**

**break;**

**case 20:**

**quicksort(B,0,m-1);//Calling quicksort function**

**qdisplay(B,m);// to ask and display number of toppers user want.**

**break;**

**case 30:**

**cout<<"the end";**

**break;**

**default:**

**cout<<"wrong choice";**

**}**

**}while(x!=30);**

**break;**

**case 6:**

**search(B,m);//Calling search function for linear searching.**

**binsearch(B,m);**

**break;**

**case 7:**

**binsearch(B,m);//Calling binsearch function for binary searching.**

**break;**

**case 8:**

**cout<<"the end";**

**break;**

**default:**

**cout<<"wrong choice";**

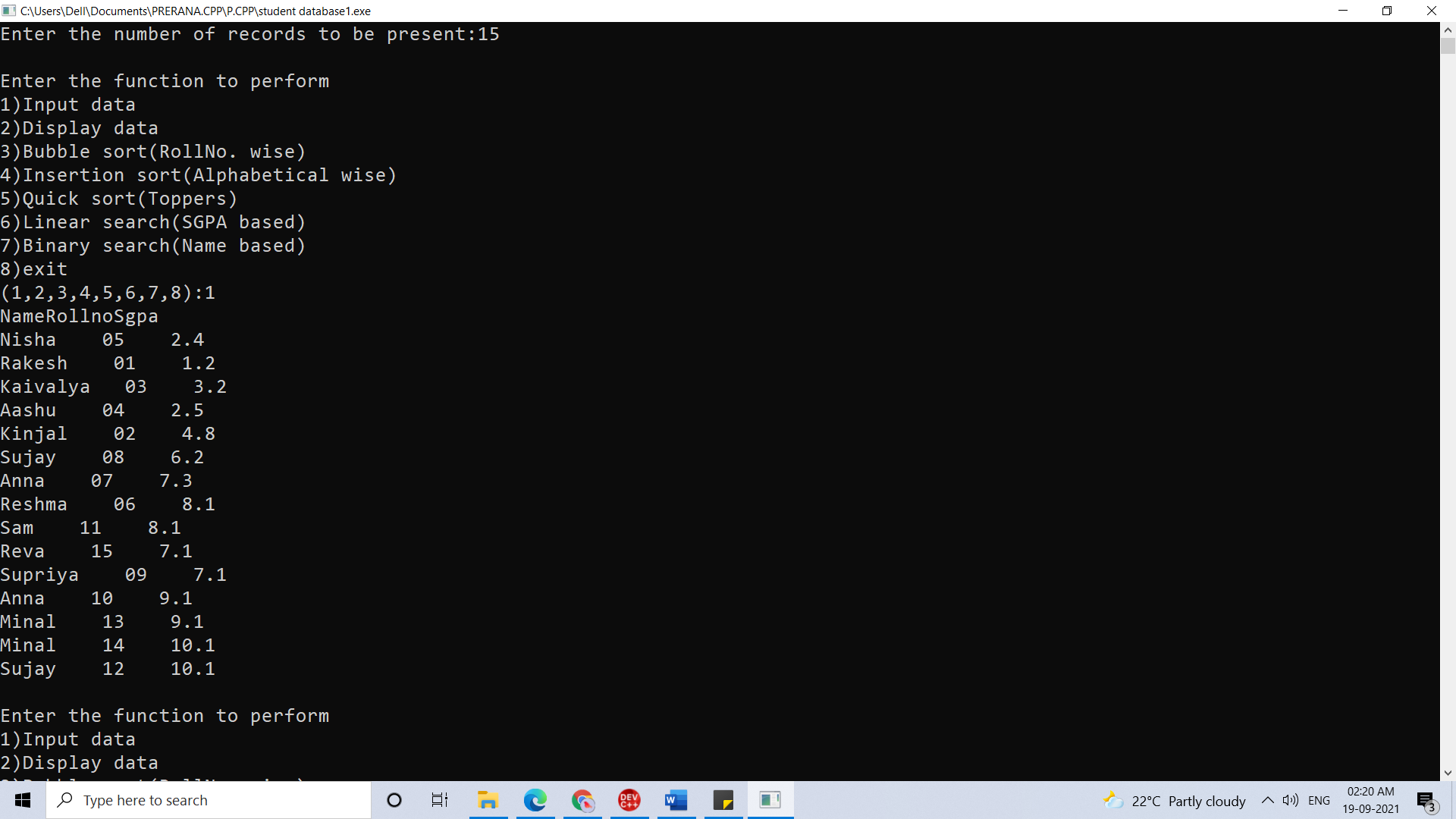
**}**

**}while(f!=8);**

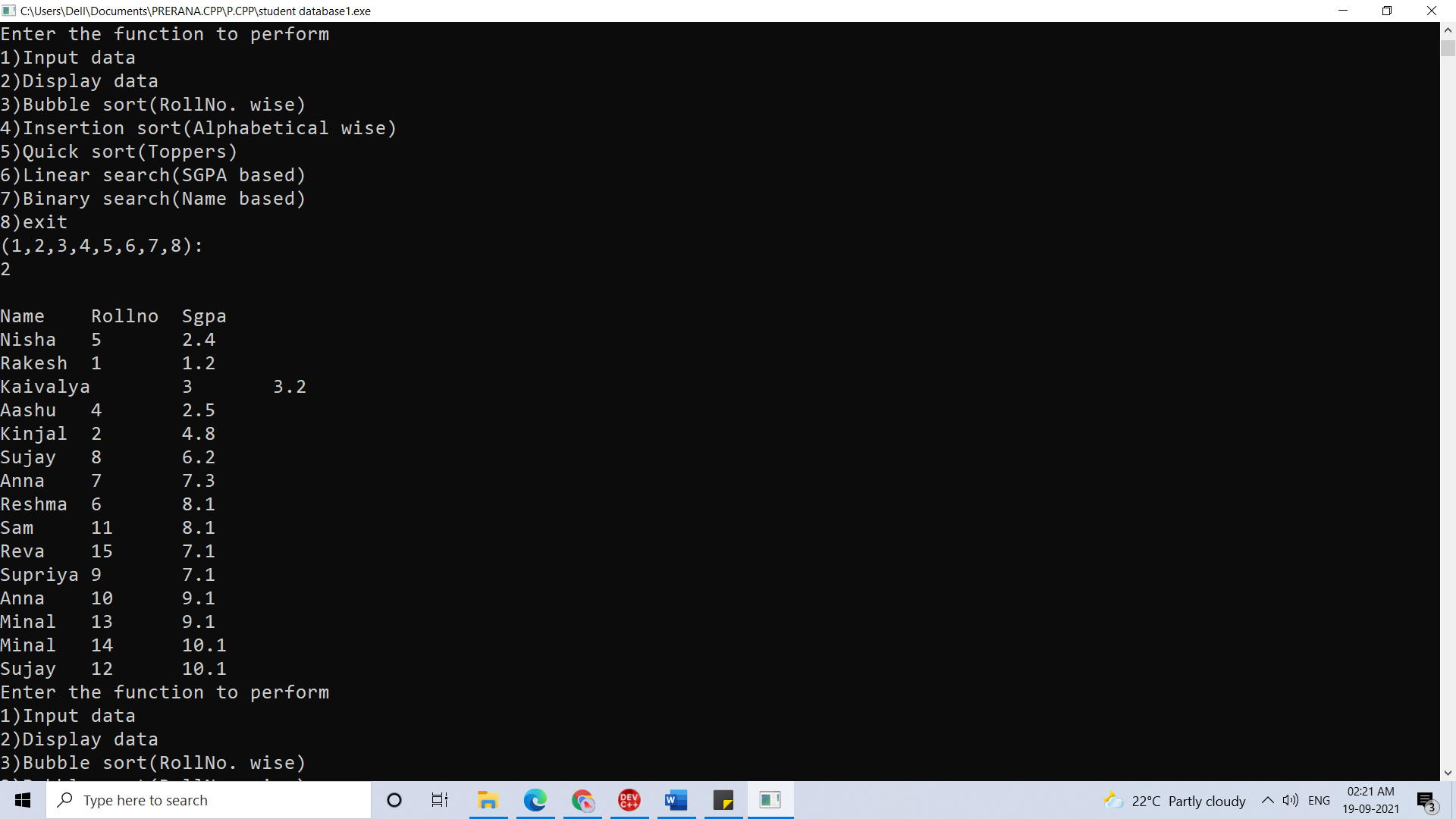
**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

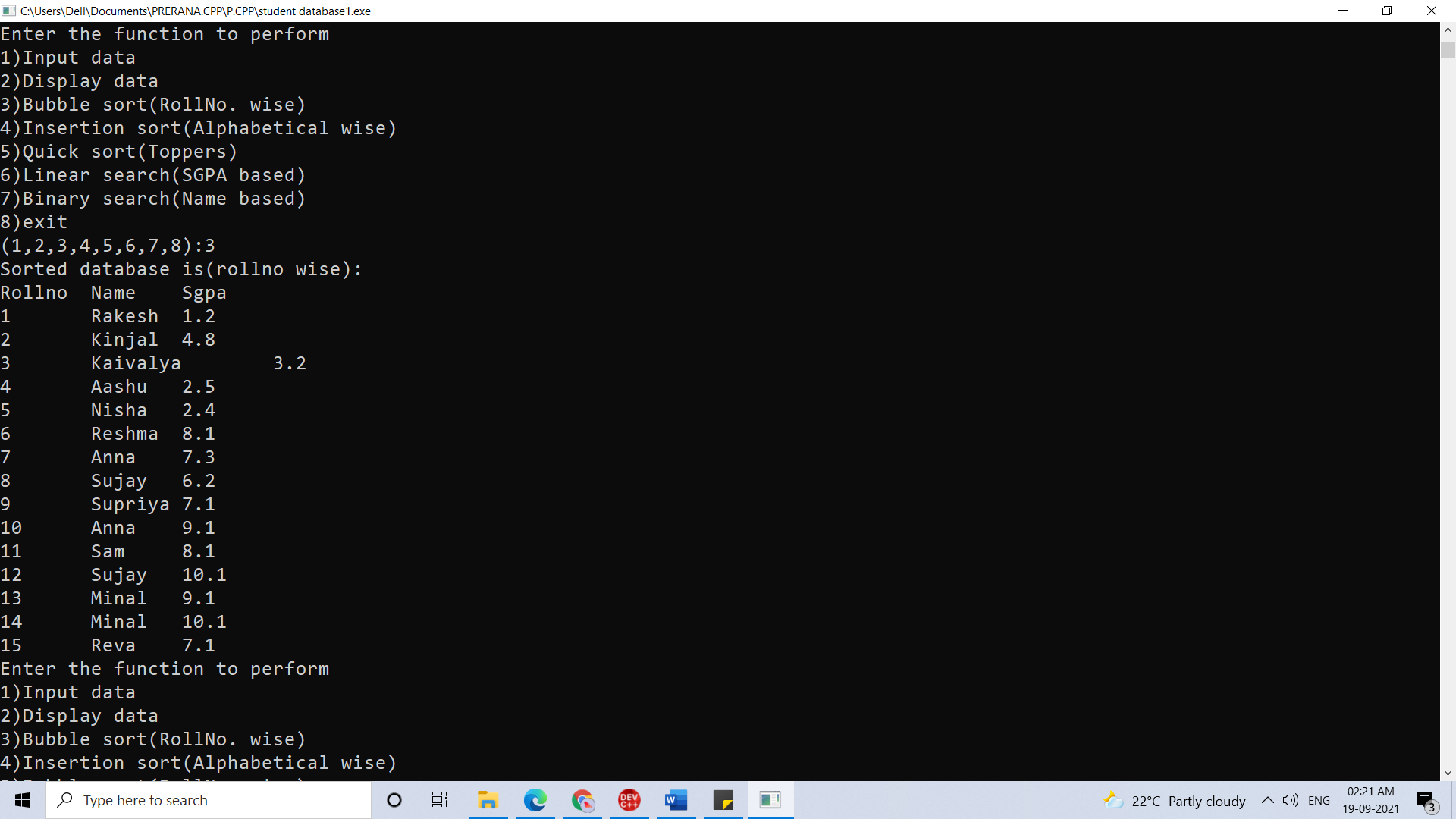
**1)Input data**



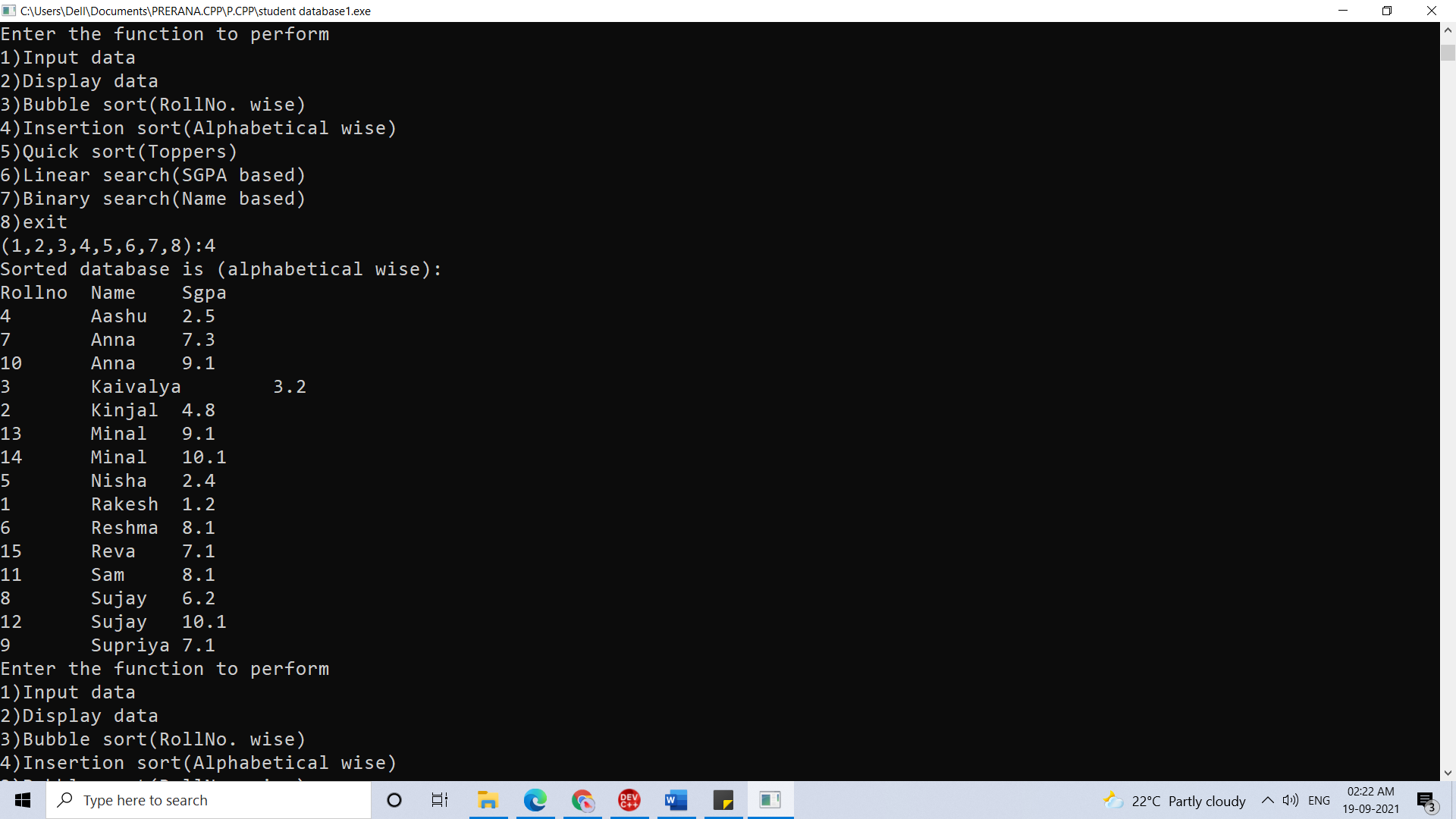
**2)Display data**



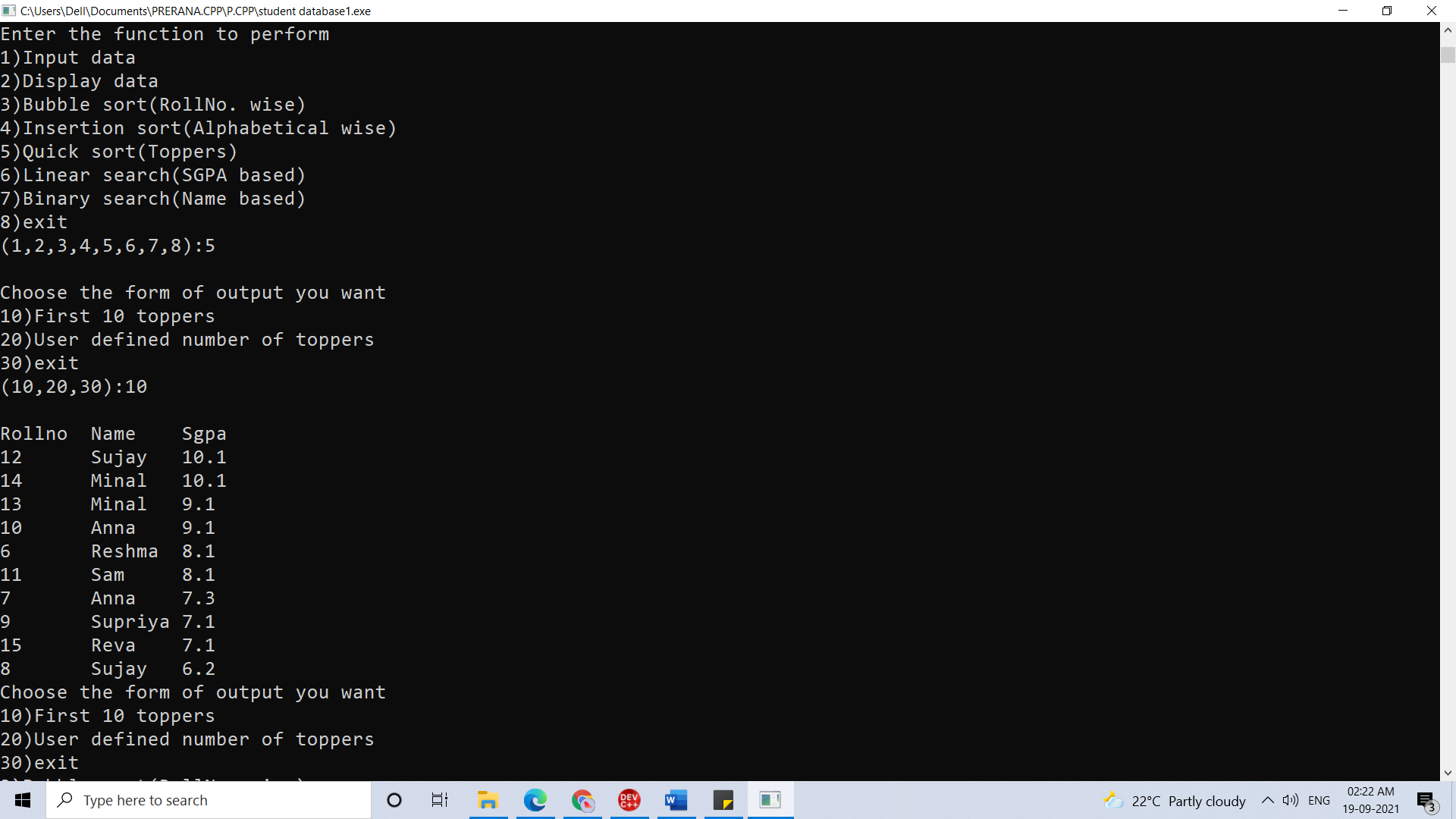
**3)Bubble sort(Roll No wise)**



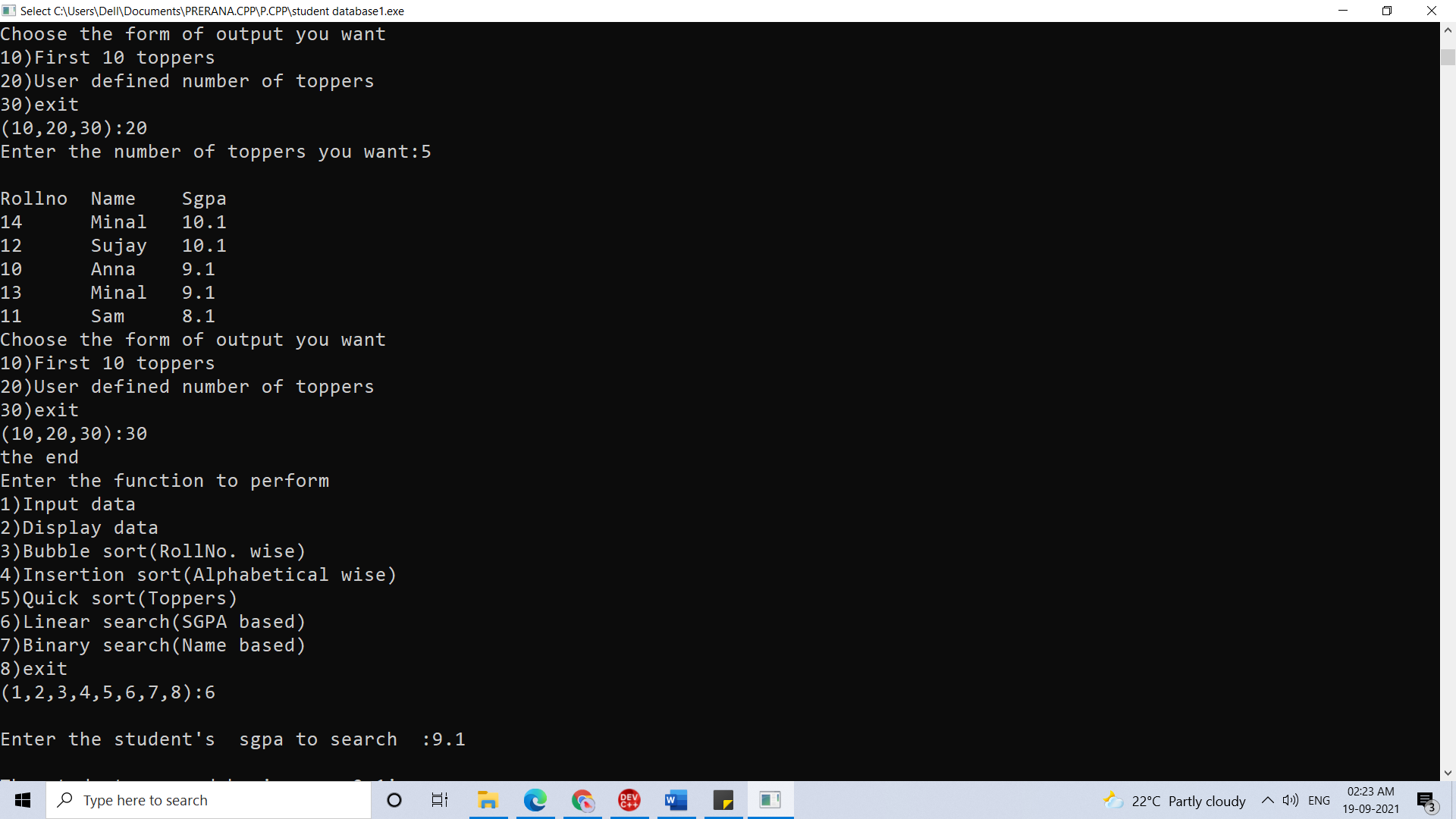
**4)Insertion sort(Alphabetical wise)**



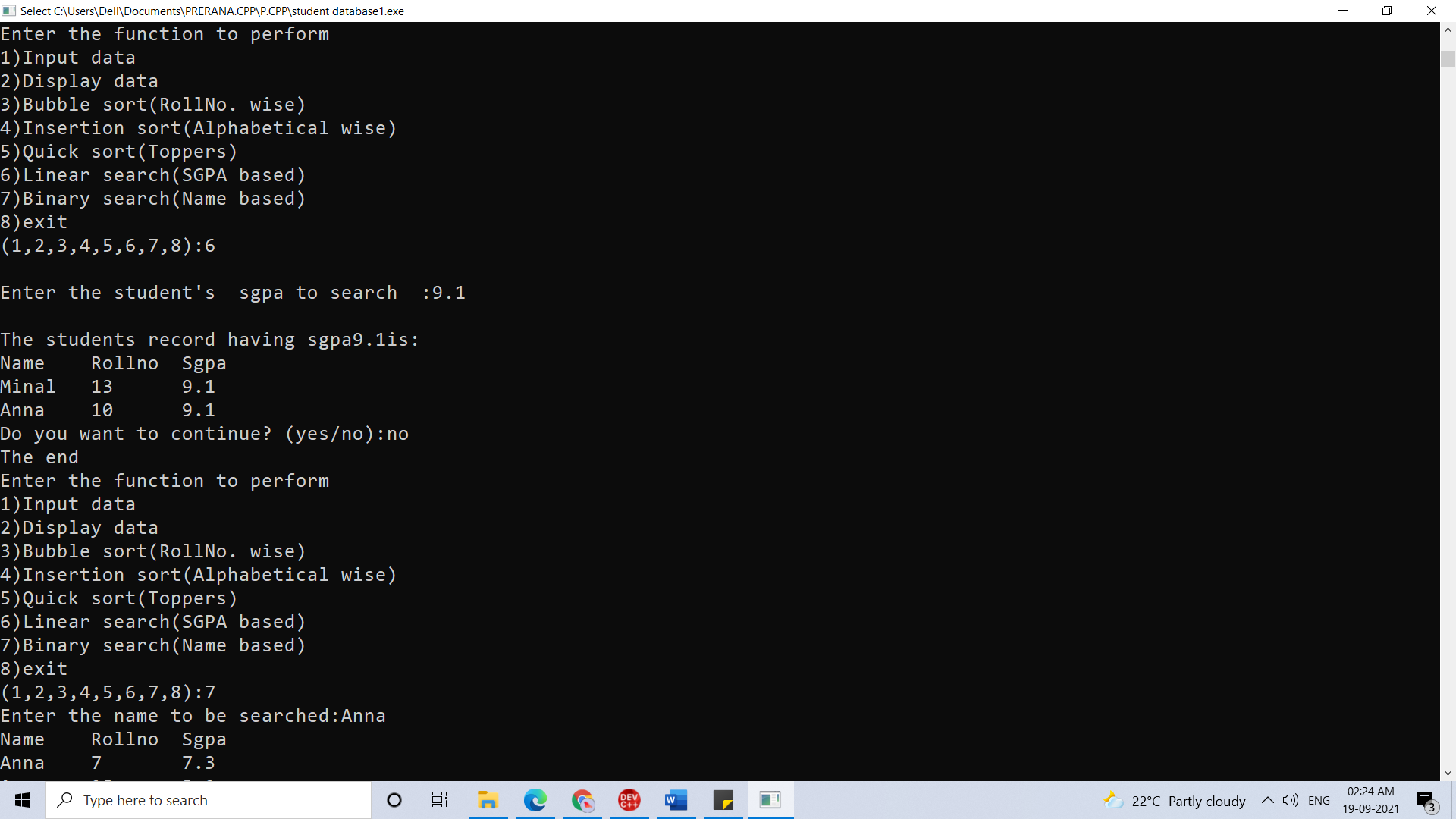
**5) i] First ten toppers**



**5)ii] User defined number of toppers**



**6)Liner search (SGPA based)**



**7)Binary search (Name based)**

