



PIMPRI CHINCHWAD EDUCATION TRUST'S.  
**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING**  
(An Autonomous Institute)

**S.Y. B. TECH**

**Year: 2024 – 25**

**Semester: I**

**Name:** Abhishek Joshi

**PRN:** 123B1B150

**Department:** Computer Engineering

**Division:** C (C1)

**Course:** Data Structures Laboratory

**Course Code:** BCE23PC02

**Date:**

## Assignment –1

- **Aim:**

Consider a student database of SY COMP class (at least 15 records). Database contains different fields of every student like Roll No, Name and SGPA.

- a. Design a roll call list, arrange list of students according to roll numbers in ascending order using Insertion Sort
- b. Arrange list of students alphabetically using shell sort
- c. Arrange list of students to find out first ten toppers from a class using Radix sort

- **Source Code :**

```
#include<iostream>
using namespace std;

class Student{
private:
string name;
int roll;
public:
void input(Student s[], int size){
for(int i=0; i<size; ++i){
cout<<"Enter Data of student - "<<(i+1)<<": ";
cin>>s[i].roll>>s[i].name;
}
}

void insertionSort(Student s[], int n){
for(int i=1; i<=n; i++){
int j=i-1;
int key = s[i].roll;
while(j>=0 && s[j].roll > key){
s[j+1].roll = s[j].roll;
j--;
}
s[j+1].roll = key;
}
}
```

```

void shellSort(Student s[], int n){
    int gap=n/2;
    while(gap>=1){
        for(int j=gap; j<n; j++){
            for(int i=j-gap; i>=0;){
                if(s[i+gap].roll > s[i].roll){
                    break;
                }
                else{
                    // Swap roll
                    int temp;
                    temp = s[i+gap].roll;
                    s[i+gap].roll = s[i].roll;
                    s[i].roll = temp;

                    // Swap name
                    string tempo;
                    tempo = s[i+gap].name;
                    s[i+gap].name = s[i].name;
                    s[i].name = tempo;
                }
                i=i-gap;
            }
        }
        gap = gap/2;
    }
}

void display(Student s[], int size){
    for(int i=0; i<size; ++i){
        cout<<"Data of student - "<<(i+1)<<": "<<s[i].roll<<" "<<s[i].name<<endl;
    }
}

};

int main(){
    Student s[3], x;
    int n=sizeof(s)/ sizeof(s[0]);
    int choice;
    do{
        cout<<"Enter choice: "; cin>>choice;
        switch(choice){
            case 1:
                x.input(s,n); break;
            case 2:
                x.insertionSort(s,n); break;

```

```

        case 3:
            x.display(s,n); break;
        case 4:
            x.shellSort(s,n); break;
        default:
            cout<<"Exit or write correct choice: "<<endl;
    }
    } while(choice!=5);
    return 0;
}

```

- **Screen Shot of Output :**

## Output

```

Enter choice:
1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit
1
Enter Roll No and Name of student - 1: 123B1B150Abhishek
Enter Roll No and Name of student - 2: 123B1B1ROHAN
Enter Roll No and Name of student - 3: 123B1B2RAKESH
Enter Roll No and Name of student - 4: 123B1B3ABI
Enter Roll No and Name of student - 5: 123B1B4MAYUR
Enter Roll No and Name of student - 6: 123B1B5ADITYA
Enter Roll No and Name of student - 7: 123B1B6DEV
Enter Roll No and Name of student - 8: 123B1B7MADRE
Enter Roll No and Name of student - 9: 123B1B8SOAM
Enter Roll No and Name of student - 10: 123B1B9RAHUL
Enter Roll No and Name of student - 11: 123B1B10WANG
Enter Roll No and Name of student - 12: 123B1B11TRUMP
Enter Roll No and Name of student - 13: 123B1B12OM
Enter Roll No and Name of student - 14: 123BIB13ARRAY
Enter Roll No and Name of student - 15: 123B1B14KUNAL

```

## Output

[Clear](#)

Enter choice:

1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit

2  
Students sorted by Roll No in ascending order.

Enter choice:

1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit

3  
Data of student - 1: 123 B1B150Abhishek  
Data of student - 2: 123 B1B1ROHAN  
Data of student - 3: 123 B1B2RAKESH  
Data of student - 4: 123 B1B3ABI  
Data of student - 5: 123 B1B4MAYUR  
Data of student - 6: 123 B1B5ADITYA  
Data of student - 7: 123 B1B6DEV  
Data of student - 8: 123 R1R7MANDRE

## Output

Cl

Data of student - 7: 123 B1B6DEV  
Data of student - 8: 123 B1B7MADRE  
Data of student - 9: 123 B1B8SOAM  
Data of student ● 10: 123 B1B9RAHUL  
Data of student - 11: 123 B1B10WANG  
Data of student - 12: 123 B1B11TRUMP  
Data of student - 13: 123 B1B12OM  
Data of student - 14: 123 BIB13ARRAY  
Data of student - 15: 123 B1B14KUNAL

Enter choice:

1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit

4

Students sorted alphabetically by Name.

Enter choice:

1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)

## Output

Clear

```
Data of student - 13: 123 B1B12OM
Data of student - 14: 123 BIB13ARRAY
Data of student - 15: 123 B1B14KUNAL

Enter choice:
1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit
4
Students sorted alphabetically by Name.

Enter choice:
1. Input Students
2. Sort by Roll No (Insertion Sort)
3. Display Students
4. Sort Alphabetically by Name (Shell Sort)
5. Exit
5
Exiting...

=== Code Execution Successful ===
```

- **Conclusion:**

Hence, we studied about various sorting techniques such as Insertion Sort, Shell Sort and Radix Sort with their algorithm and programs.