**CODE:-**

#include <iostream>

#include <fstream>

using namespace std;

void bubbleSort(int array[], int size)

{

for (int step = 0; step < size - 1; step++)

{

for (int i = 0; i < size - step - 1; i++)

{

if (array[i] > array[i + 1])

{

int temp = array[i];

array[i] = array[i + 1];

array[i + 1] = temp;

}

}

}

}

int main()

{

int num;

char c;

fstream f1, f2, f3;

f1.open("file1.txt", ios::out);

cout << "Enter the elements into the first file." << endl;

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

{

cin >> num;

f1 << num << " ";

}

f1.close();

f1.open("file1.txt", ios::in);

f2.open("file2.txt", ios::out);

cout << "Enter the elements into the second file." << endl;

for (int i = 0; i < 3; i++)

{

cin >> num;

f2 << num << " ";

}

f2.close();

f2.open("file2.txt", ios::in);

f3.open("file3.txt", ios::out);

cout << "Contents of 1st file." << endl;

while (f1)

{

f1 >> num;

if (f1.eof() == 1)

continue;

cout << num << " ";

f3 << num << " ";

}

cout << endl

<< "Contents of 2nd file." << endl;

while (f2)

{

f2 >> num;

if (f2.eof() == 1)

continue;

cout << num << " ";

f3 << num << " ";

}

f1.close();

f2.close();

f3.close();

f3.open("file3.txt", ios::in);

int a[8], i = 0;

while (f3)

{

f3 >> a[i++];

if (f3.eof() == 1)

continue;

}

f3.close();

bubbleSort(a, 8);

f3.open("file3.txt", ios::out);

for (int i = 0; i < 8; i++)

{

f3 << a[i] << " ";

}

f3.close();

f3.open("file3.txt", ios::in);

cout << endl

<< "Contents of 3rd file." << endl;

while (f3)

{

f3 >> num;

if (f3.eof() == 1)

continue;

cout << num << " ";

}

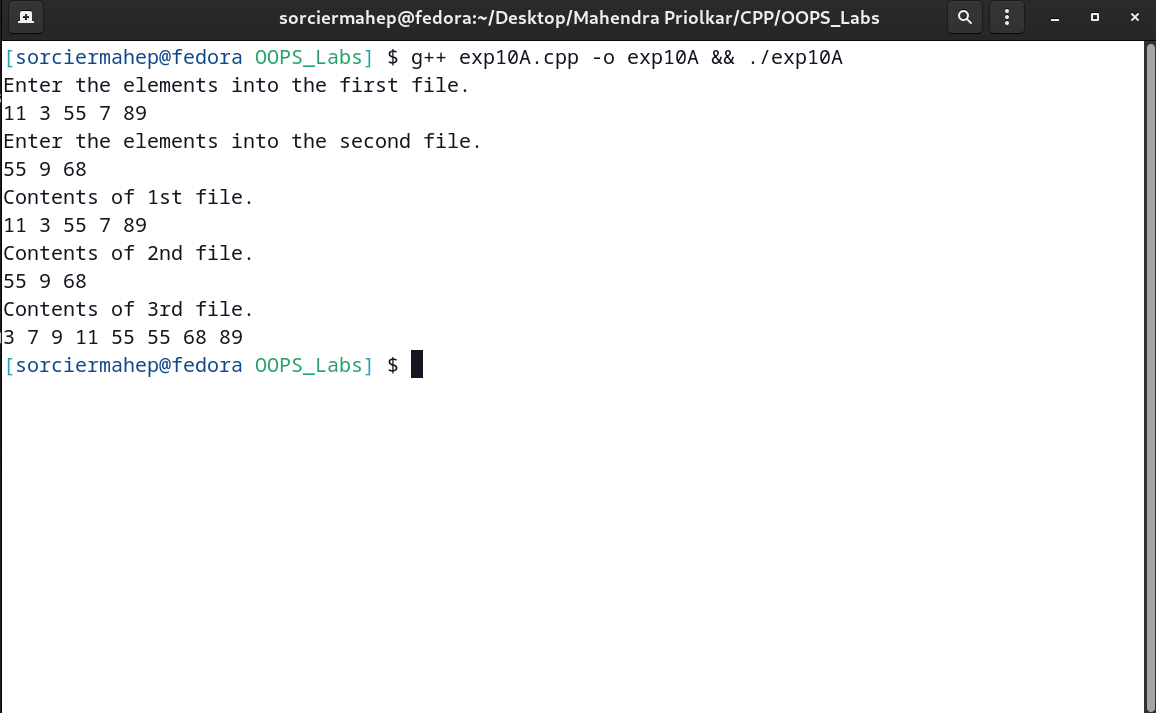
cout << endl;

f3.close();

return 0;

}

**OUTPUT:-**



**CODE:-**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

class telephone

{

char name[20];

int phone;

public:

void getname()

{

cout << "Enter Name: " << endl;

getchar();

cin.getline(name, 20);

}

void getphone()

{

cout << "Enter Phone Number: " << endl;

cin >> phone;

}

void show()

{

cout << "Name: " << name << endl;

cout << "Phone Number: " << phone << endl;

}

char \*ret\_name()

{

return name;

}

};

void add\_telephone()

{

telephone t1;

fstream f1;

f1.open("telephone.txt", ios::binary | ios::app);

t1.getname();

t1.getphone();

f1.write((char \*)&t1, sizeof(t1));

f1.close();

cout << "Telephone Directory Has Been Created" << endl;

cin.get();

}

void display\_records()

{

telephone t1;

fstream f1;

f1.open("telephone.txt", ios::binary | ios::in);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

cout << "The records are: " << endl;

while (f1.read((char \*)&t1, sizeof(t1)))

{

t1.show();

cout << endl;

}

f1.close();

}

void display\_phone(char \*str)

{

telephone t1;

fstream f1;

f1.open("telephone.txt", ios::binary | ios::in);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

int flag = 0;

while (f1.read((char \*)&t1, sizeof(t1)))

{

if (strcmp(t1.ret\_name(), str) == 0)

{

t1.show();

flag = 1;

}

}

f1.close();

if (flag == 0)

cout << "Record does not exist." << endl;

}

void delete\_telephone(char \*str)

{

telephone t1;

fstream f1;

f1.open("telephone.txt", ios::binary | ios::in | ios::out);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

fstream f2;

f2.open("temp.txt", ios::out);

f1.seekg(0, ios::beg);

while (f1.read((char \*)&t1, sizeof(t1)))

{

if (strcmp(t1.ret\_name(), str) != 0)

{

f2.write((char \*)&t1, sizeof(t1));

}

}

f1.close();

f2.close();

remove("telephone.txt");

rename("temp.txt", "telephone.txt");

cout << "Record successfully deleted." << endl;

}

void modify\_telephone(char \*str)

{

int flag = 0;

telephone t1;

fstream f1;

f1.open("telephone.txt", ios::binary | ios::in | ios::out);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

while (!f1.eof() && flag == 0)

{

f1.read((char \*)&t1, sizeof(t1));

if (strcmp(t1.ret\_name(), str) == 0)

{

t1.show();

cout << "Enter The New Details." << endl;

t1.getphone();

int pos = (-1) \* (int)(sizeof(t1));

f1.seekp(pos, ios::cur);

f1.write((char \*)&t1, sizeof(t1));

cout << "Successfully updated." << endl;

flag = 1;

}

}

f1.close();

if (flag == 0)

cout << "Record does not exist." << endl;

}

int main()

{

int ch;

char str[20];

while (1)

{

cout << "1.Add Telephone Records." << endl;

cout << "2.Show Telephone Records." << endl;

cout << "3.Search a Telephone Record as per name." << endl;

cout << "4.Modify a Telephone Record as per name." << endl;

cout << "5.Delete a Telephone Record as per name." << endl;

cout << "6.Exit." << endl;

cin >> ch;

switch (ch)

{

case 1:

add\_telephone();

break;

case 2:

display\_records();

break;

case 3:

cout << "Enter the name." << endl;

cin >> str;

display\_phone(str);

break;

case 4:

cout << "Enter the name." << endl;

cin >> str;

modify\_telephone(str);

break;

case 5:

cout << "Enter the name." << endl;

cin >> str;

delete\_telephone(str);

break;

case 6:

exit(1);

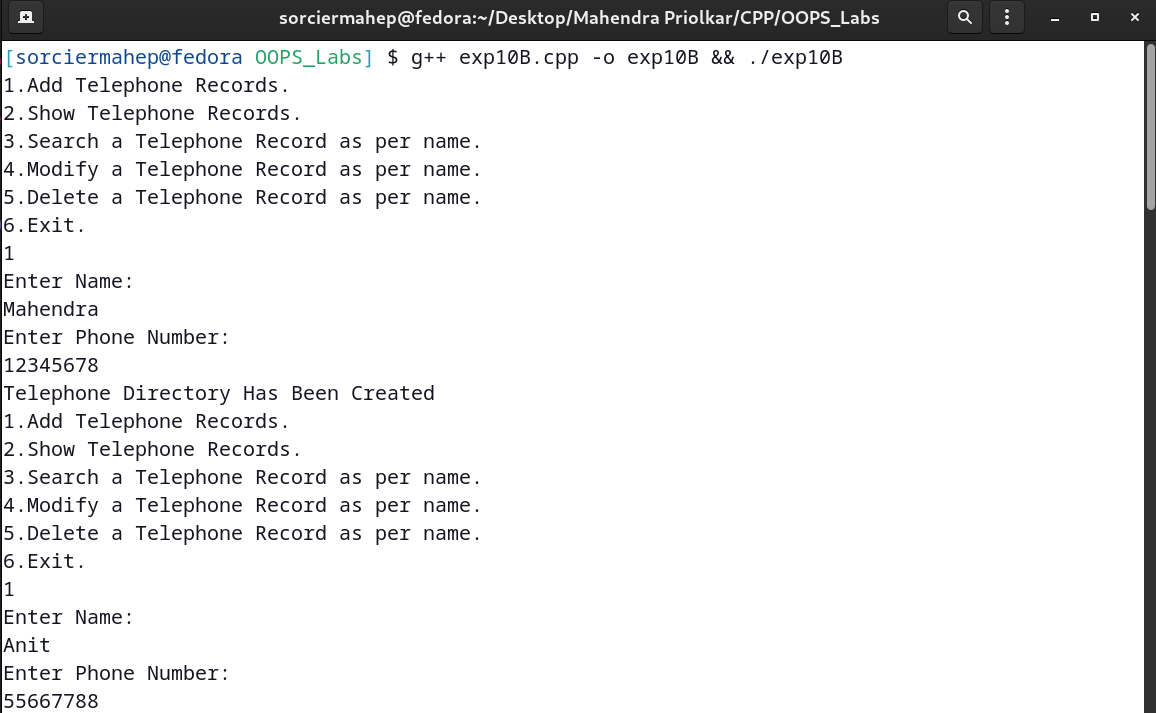
default:

cout << "Erroneous input." << endl;

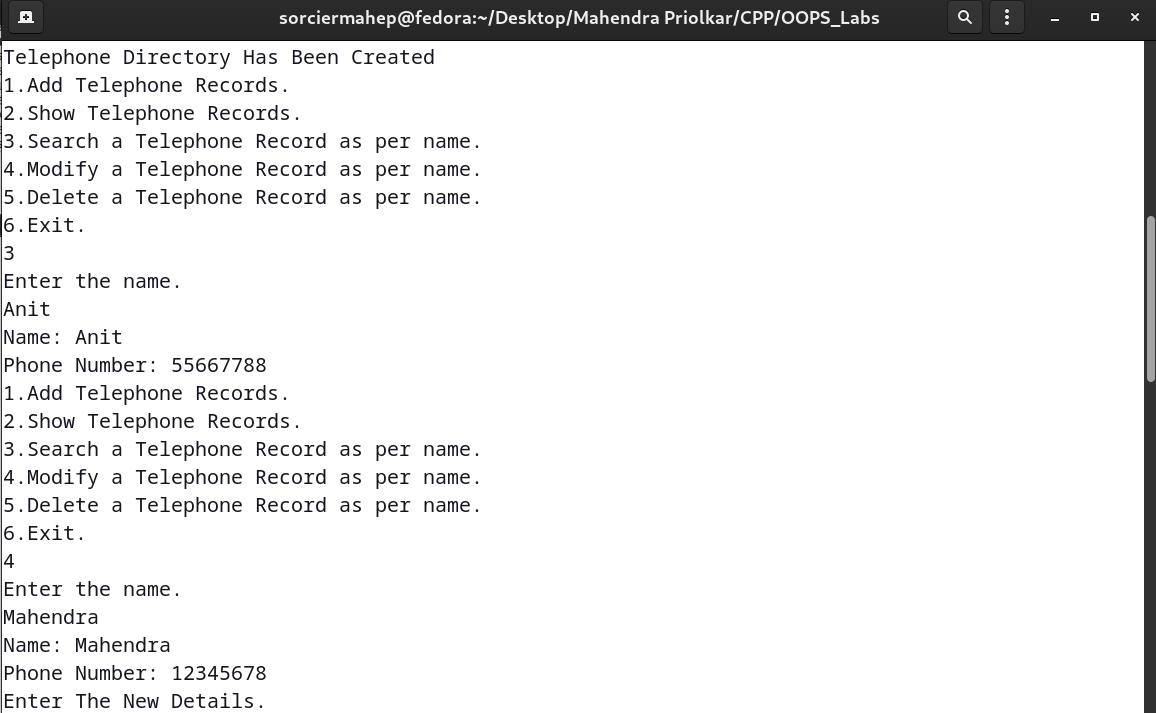
}

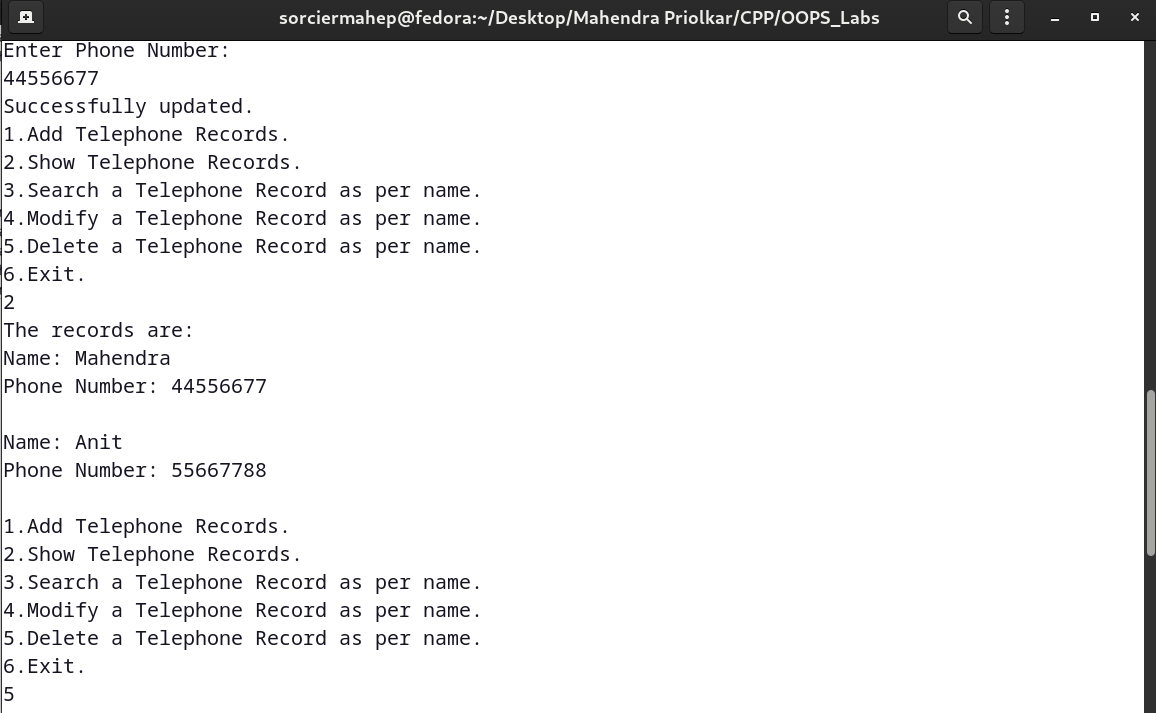
}

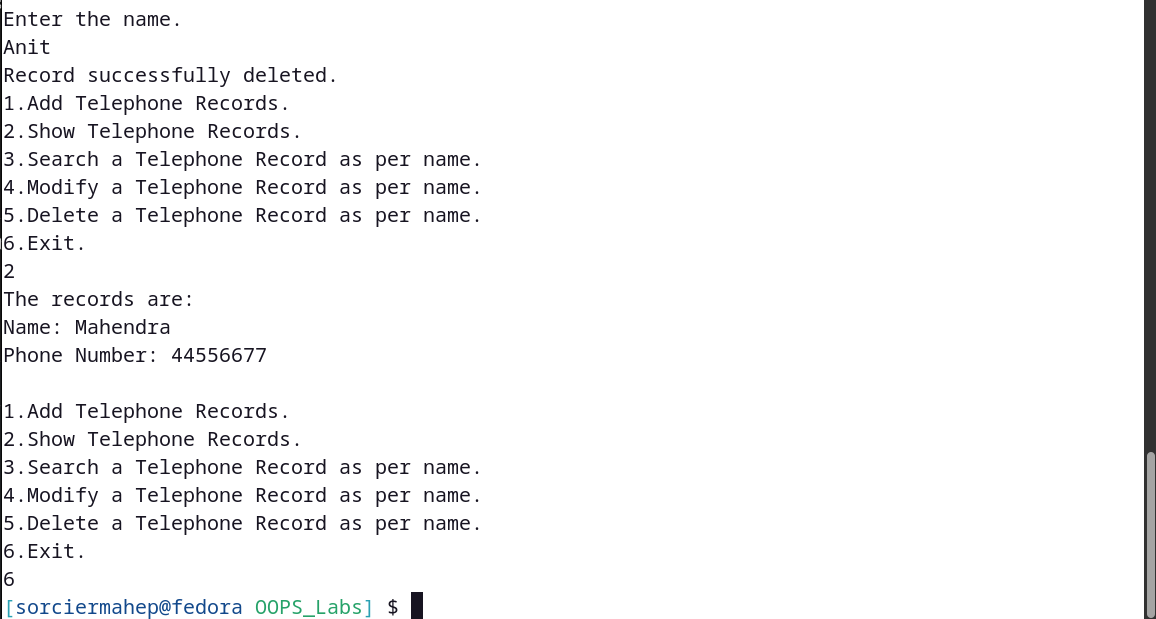
return 0;

}

**OUTPUT:-**







**CODE:-**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

class student

{

char name[20];

int rollno;

char address[30];

char branch[10];

public:

void getroll()

{

cout << "Enter roll number." << endl;

cin >> rollno;

}

void getdetails()

{

cout << "Enter name." << endl;

getchar();

cin.getline(name, 20);

cout << "Enter address." << endl;

cin.getline(address, 30);

cout << "Enter branch." << endl;

cin.getline(branch, 10);

}

void show()

{

cout << "Name: " << name << endl;

cout << "Address: " << address << endl;

cout << "Roll number: " << rollno << endl;

cout << "Branch: " << branch << endl;

}

int rollret()

{

return rollno;

}

char \*retname()

{

return name;

}

};

void addstudent(student s[], int n)

{

fstream f1;

char \*s2, temp[20];

s[n].getroll();

s[n].getdetails();

strcpy(temp, s[n].retname());

s2 = strcat(temp, ".txt");

f1.open(s2, ios::binary | ios::app);

f1.write((char \*)&s[n], sizeof(s[n]));

f1.close();

cout << "Student Database Has Been Created." << endl;

cin.get();

}

void display\_records(char \*str)

{

fstream f1;

student s1;

char \*s2, temp[20];

strcpy(temp, str);

s2 = strcat(temp, ".txt");

f1.open(s2, ios::binary | ios::in);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

cout << "The records are: " << endl;

while (f1.read((char \*)&s1, sizeof(s1)))

{

s1.show();

cout << endl;

}

f1.close();

}

int search\_roll(char \*str, int roll)

{

student s1;

fstream f1;

char \*s2, temp[20];

strcpy(temp, str);

s2 = strcat(temp, ".txt");

f1.open(s2, ios::binary | ios::in);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return -1;

}

while (f1.read((char \*)&s1, sizeof(s1)))

{

if (s1.rollret() == roll)

{

s1.show();

return 1;

}

}

f1.close();

return 0;

}

void delete\_details(char \*str, int roll)

{

student s1;

fstream f1;

char \*s2, temp[20];

strcpy(temp, str);

s2 = strcat(temp, ".txt");

f1.open(s2, ios::binary | ios::in);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return;

}

fstream f2;

f2.open("temp.txt", ios::binary | ios::out);

f1.seekg(0, ios::beg);

while (f1.read((char \*)&s1, sizeof(s1)))

{

if (s1.rollret() != roll)

{

f2.write((char \*)&s1, sizeof(s1));

}

}

f1.close();

f2.close();

remove(s2);

rename("temp.txt", s2);

f1.open(s2, ios::binary | ios::in);

f1.seekg(0, ios::end);

int file\_size = f1.tellg();

if (file\_size == 0)

remove(s2);

cout << "Record successfully deleted." << endl;

}

int modify\_details(char \*str, int roll)

{

student s1;

fstream f1;

char \*s2, temp[20];

strcpy(temp, str);

s2 = strcat(temp, ".txt");

f1.open(s2, ios::binary | ios::in | ios::out);

if (!f1.good())

{

cout << "File couldn't be opened." << endl;

cin.get();

return -1;

}

while (f1.read((char \*)&s1, sizeof(s1)))

{

if (s1.rollret() == roll)

{

s1.show();

cout << "Enter The New Details." << endl;

s1.getdetails();

int pos = (-1) \* (int)(sizeof(s1));

f1.seekp(pos, ios::cur);

f1.write((char \*)&s1, sizeof(s1));

cout << "Successfully updated." << endl;

return 1;

}

}

f1.close();

return 0;

}

int main()

{

int ch, roll, i = 0, check, flag = 0;

student s[10];

while (1)

{

flag = 0;

cout << "1.Add details." << endl;

cout << "2.Display details." << endl;

cout << "3.Search for student using roll number." << endl;

cout << "4.Update student details using roll number." << endl;

cout << "5.Delete student details using roll number." << endl;

cout << "6.Exit." << endl;

cin >> ch;

switch (ch)

{

case 1:

addstudent(s, i++);

break;

case 2:

cout << "The details are: " << endl;

for (int j = 0; j < i; j++)

display\_records(s[j].retname());

break;

case 3:

cout << "Enter roll number." << endl;

cin >> roll;

for (int j = 0; j < i; j++)

check = search\_roll(s[j].retname(), roll);

if (!check)

cout << "Records do not exist." << endl;

break;

case 4:

cout << "Enter roll number." << endl;

cin >> roll;

for (int j = 0; j < i; j++)

check = modify\_details(s[j].retname(), roll);

if (!check)

cout << "Records do not exist." << endl;

break;

case 5:

cout << "Enter roll number." << endl;

cin >> roll;

for (int j = 0; j < i; j++)

{

if (s[j].rollret() == roll)

{

flag = 1;

for (int k = j; k < i - 1; k++)

{

s[k] = s[k + 1];

}

delete\_details(s[j].retname(), roll);

}

}

if (flag)

i--;

else

cout << "Records do not exist." << endl;

break;

case 6:

exit(1);

default:

cout << "Erroneous input." << endl;

}

}

}

**OUTPUT:-**

