**CODE:-**

#include <iostream>

using namespace std;

class Integer

{

int i;

public:

Integer(int m=0)

{

i = m;

}

Integer operator++() // Unary Prefix Increment

{

Integer temp;

temp.i = ++i;

return temp;

}

Integer operator++(int) // Unary Postfix Increment

{

Integer temp;

temp.i = i++;

return temp;

}

Integer operator--() // Unary Prefix Decrement

{

Integer temp;

temp.i = --i;

return temp;

}

Integer operator--(int) // Unary Postfix Decrement

{

Integer temp;

temp.i = i--;

return temp;

}

void display()

{

cout << "The value is " << i << endl;

}

};

int main()

{

Integer I1(5), I2;

cout << "Initial value." << endl;

I1.display();

I2 = ++I1;

cout << "Value after pre increment." << endl;

I2.display();

I2 = I1++;

cout << "Value after post increment." << endl;

I2.display();

cout << "Initial value." << endl;

I1.display();

I2 = --I1;

cout << "Value after pre decrement." << endl;

I2.display();

I2 = I1--;

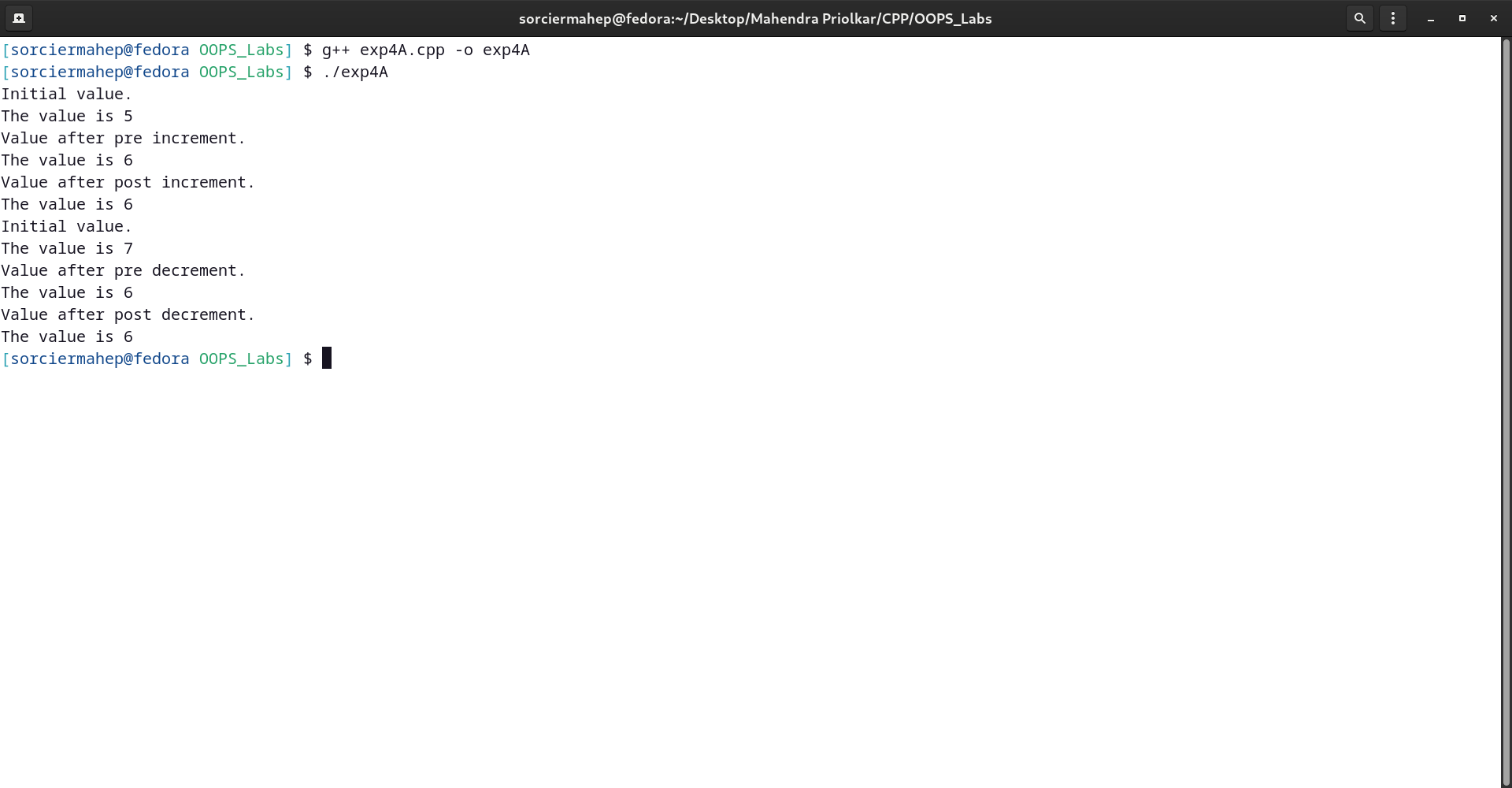
cout << "Value after post decrement." << endl;

I2.display();

return 0;

}

**OUTPUT:-**



**CODE:-**

#include <iostream>

using namespace std;

class Integer

{

int m, n;

public:

Integer(int a = 0, int b = 0)

{

m = a;

n = b;

}

Integer operator\*(Integer &a)

{

Integer temp;

temp.m = m \* a.m;

temp.n = n \* a.n;

return temp;

}

Integer operator-(int a)

{

Integer temp;

temp.m = m - a;

temp.n = n - a;

return temp;

}

void display()

{

cout << "Values of m and n are " << m << " and " << n << endl;

}

friend Integer operator+(int, Integer);

friend int main();

};

Integer operator+(int a, Integer I)

{

Integer temp;

temp.m = a + I.m;

temp.n = a + I.n;

return temp;

}

int main()

{

Integer x, y(5, 6);

cout << "x=5+y" << endl;

cout << "Values of y: " << endl;

y.display();

x = 5 + y;

cout << "Values of x:" << endl;

x.display();

cout << "x=x\*y" << endl;

x = Integer(3, 4);

y = Integer(5, 6);

cout << "Values of x: " << endl;

x.display();

cout << "Values of y: " << endl;

y.display();

x = x \* y;

cout << "New values of x: " << endl;

x.display();

x = Integer();

y = Integer(5, 6);

cout << "x=y-5" << endl;

cout << "Values of y: " << endl;

y.display();

x = y - 5;

cout << "Values of x:" << endl;

x.display();

}

**OUTPUT:-**



**CODE:-**

#include <iostream>

using namespace std;

class Integer

{

int m, n;

public:

Integer(int a = 0, int b = 0)

{

m = a;

n = b;

}

friend ostream &operator<<(ostream &, const Integer &);

friend istream &operator>>(istream &, Integer &);

};

ostream &operator<<(ostream &os, const Integer &I)

{

os << "m:" << I.m << " "

<< "n:" << I.n << endl;

return os;

}

istream &operator>>(istream &is, Integer &I)

{

is >> I.m >> I.n;

return is;

}

int main()

{

Integer I1, I2;

cout << "Enter values of I1 and I2." << endl;

cin >> I1 >> I2;

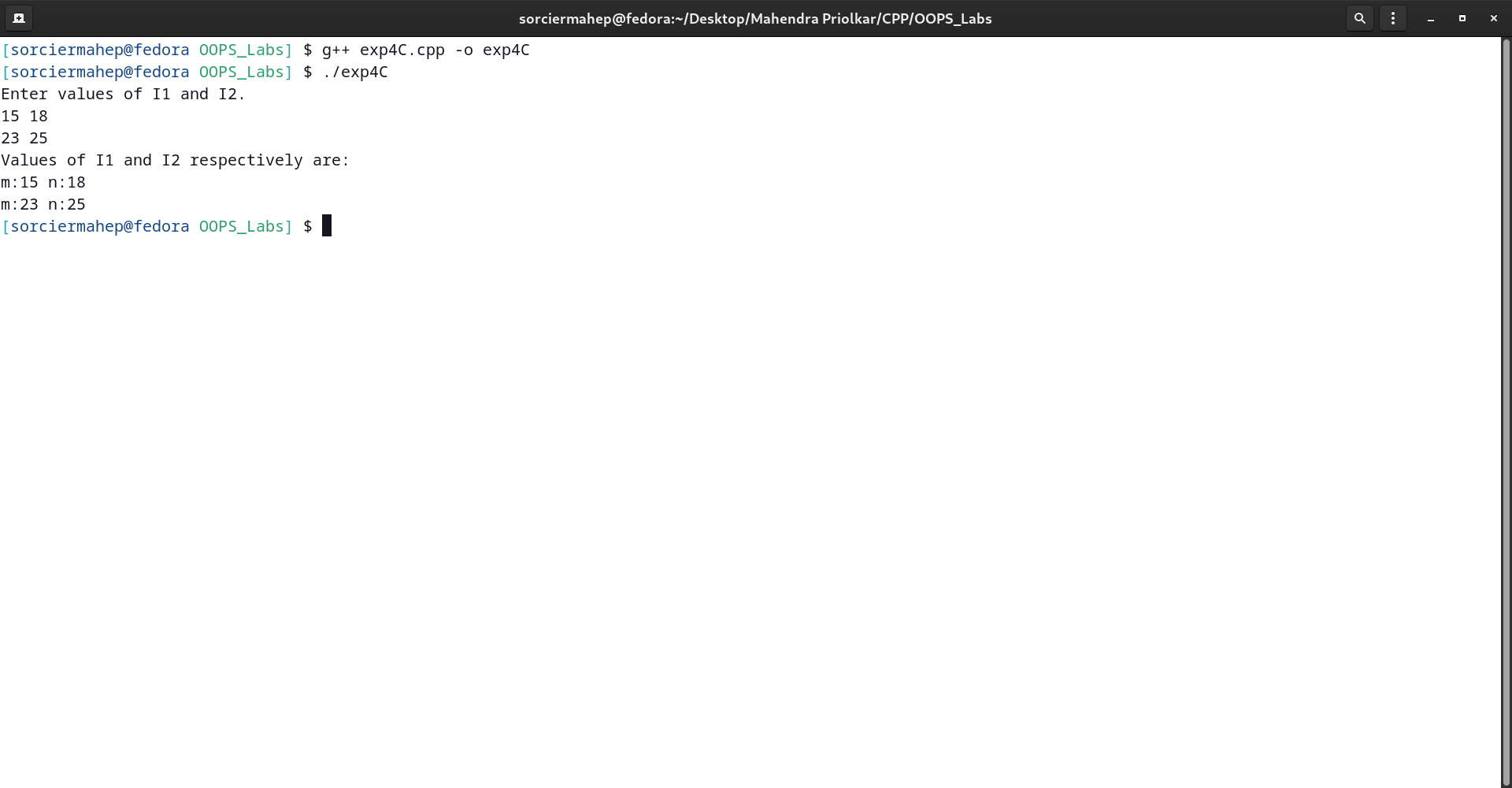
cout << "Values of I1 and I2 respectively are:" << endl;

cout << I1 << I2;

return 0;

}

**OUTPUT:-**



**CODE:-**

#include <iostream>

#include <cstring>

using namespace std;

class String

{

char \*p;

int len;

public:

String()

{

len = 0;

p = NULL;

}

String(char \*s)

{

len = strlen(s);

p = new char[len + 1];

strcpy(p, s);

}

String operator+(String &s)

{

String temp;

temp.len = len + s.len;

temp.p = new char[temp.len + 1];

strcpy(temp.p, p);

strcat(temp.p, s.p);

return temp;

}

int operator<(String &s)

{

int m = strlen(p);

int n = strlen(s.p);

if (m < n)

return 1;

else

return 0;

}

void display()

{

cout << p << endl;

}

};

int main()

{

char s2[20], s3[20];

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

cin >> s2 >> s3;

String str1, str2, str3;

str2 = String(s2);

str3 = String(s3);

str1 = str2 + str3;

str1.display();

char s4[20], s5[20];

cout << "Enter the strings to be compared." << endl;

cin >> s4 >> s5;

String str4, str5;

str4 = String(s4);

str5 = String(s5);

if (str4 < str5)

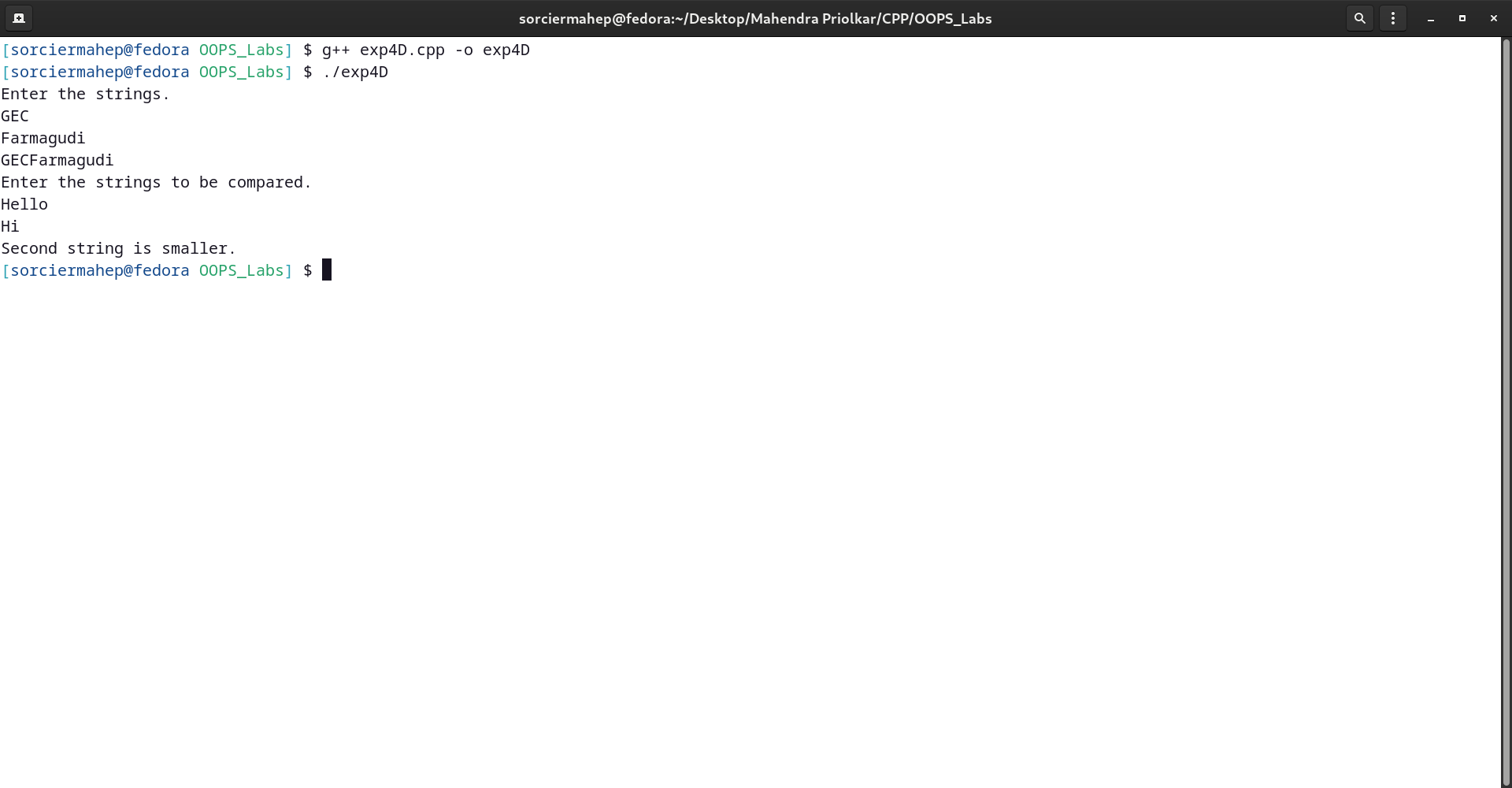
cout << "First string is smaller." << endl;

else

cout << "Second string is smaller." << endl;

}

**OUTPUT:-**



**CODE:-**

#include <iostream>

using namespace std;

class vector

{

int n;

public:

void getsize()

{

cout << "Enter size." << endl;

cin >> n;

}

private:

int \*A = new int(n);

public:

void getelem()

{

cout << "Enter the elements of the vector." << endl;

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

cin >> \*(A + i);

}

void display()

{

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

cout << \*(A + i) << " ";

cout << endl;

}

vector operator+(vector &v)

{

vector w;

w.n = n;

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

{

w.A[i] = A[i] + v.A[i];

}

return w;

}

friend vector operator\*(int &, vector &);

};

vector operator\*(int &m, vector &v)

{

vector w;

w.n = v.n;

for (int i = 0; i < v.n; i++)

{

w.A[i] = m \* v.A[i];

}

return w;

}

int main()

{

vector v1, v2;

int m;

v1.getsize();

v1.getelem();

cout << "Original values." << endl;

v1.display();

cout << "Enter value of m." << endl;

cin >> m;

v2 = m \* v1;

cout << "Changed values." << endl;

v2.display();

vector v3, v4, v5;

cout << "Enter the vectors for addition.Size should be same." << endl;

v3.getsize();

v3.getelem();

cout << "Original values." << endl;

v3.display();

v4.getsize();

v4.getelem();

cout << "Original values." << endl;

v4.display();

v5 = v3 + v4;

cout << "Added vector." << endl;

v5.display();

return 0;

}

**OUTPUT:-**

