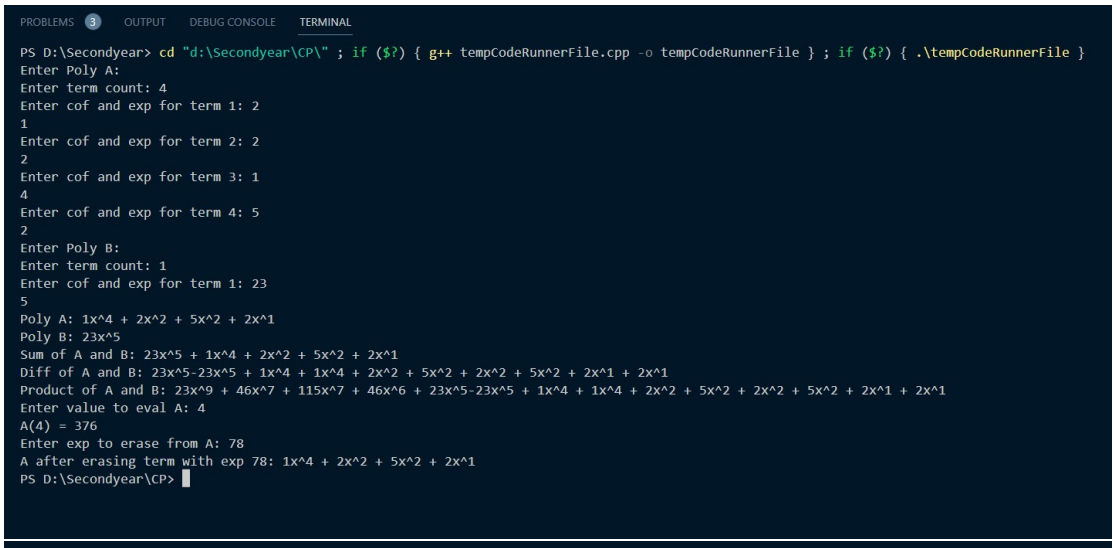
Name = Antas jain
Roll = BT22CSH015
DSA assignment



```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Secondyear> cd "d:\Secondyear\CP\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter Poly A:
Enter term count: 4
Enter cof and exp for term 1: 2
1
Enter cof and exp for term 2: 2
2
Enter cof and exp for term 3: 1
4
Enter cof and exp for term 4: 5
2
Enter Poly B:
Enter term count: 1
Enter cof and exp for term 1: 23
5
Poly A: 1x^4 + 2x^2 + 5x^2 + 2x^1
Poly B: 23x^5
Sum of A and B: 23x^5 + 1x^4 + 2x^2 + 5x^2 + 2x^1
Diff of A and B: 23x^5-23x^5 + 1x^4 + 1x^4 + 2x^2 + 5x^2 + 2x^2 + 5x^2 + 2x^1 + 2x^1
Product of A and B: 23x^9 + 46x^7 + 115x^7 + 46x^6 + 23x^5-23x^5 + 1x^4 + 1x^4 + 2x^2 + 5x^2 + 2x^2 + 5x^2 + 2x^1 + 2x^1
Enter value to eval A: 4
A(4) = 376
Enter exp to erase from A: 78
A after erasing term with exp 78: 1x^4 + 2x^2 + 5x^2 + 2x^1
PS D:\Secondyear\CP>
```

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

class Trm
{
public:
    int cof;
    int exp;
    Trm *nxt;
```

```cpp
    Trm(int c, int e) : cof(c), exp(e), nxt(nullptr) {}
};
```

```cpp
class Ply
{
private:
    Trm *hd;
```

```cpp
public:
    Ply()
    {
        hd = new Trm(0, -1);
        hd->nxt = hd;
    }
```

```cpp
    void Rd()
    {
```

```
        int n;
        cout << "Enter term count: ";
        cin >> n;

        for (int i = 0; i < n; i++)
        {
            int c, e;
            cout << "Enter cof and exp for term " << i + 1 << ": ";
            cin >> c >> e;

            Ins(c, e);
        }
    }

    void Ins(int c, int e)
    {
        Trm *nNd = new Trm(c, e);
        Trm *cur = hd;

        while (cur->nxt != hd && cur->nxt->exp >= e)
        {
            cur = cur->nxt;
        }

        nNd->nxt = cur->nxt;
        cur->nxt = nNd;
    }

    void Wr()
    {
        Trm *cur = hd->nxt;
        bool frst = true;

        while (cur != hd)
        {
            if (cur->cof != 0)
            {
                if (!frst && cur->cof > 0)
                {
                    cout << " + ";
                }

                if (cur->exp == 0)
                {
                    cout << cur->cof;
                }
                else
                {
                    cout << cur->cof << "x^" << cur->exp;
                }
```

```
                frst = false;
            }
            cur = cur->nxt;
        }
```

```
        cout << endl;
    }
```

```
    void Add(Ply &a, Ply &b)
    {
        Trm *trmA = a.hd->nxt;
        Trm *trmB = b.hd->nxt;
```

```
        while (trmA != a.hd && trmB != b.hd)
        {
            if (trmA->exp > trmB->exp)
            {
                Ins(trmA->cof, trmA->exp);
                trmA = trmA->nxt;
            }
            else if (trmA->exp < trmB->exp)
            {
                Ins(trmB->cof, trmB->exp);
                trmB = trmB->nxt;
            }
            else
            {
                int sum = trmA->cof + trmB->cof;
                if (sum != 0)
                {
                    Ins(sum, trmA->exp);
                }
                trmA = trmA->nxt;
                trmB = trmB->nxt;
            }
        }
```

```
        while (trmA != a.hd)
        {
            Ins(trmA->cof, trmA->exp);
            trmA = trmA->nxt;
        }
```

```
        while (trmB != b.hd)
        {
            Ins(trmB->cof, trmB->exp);
            trmB = trmB->nxt;
        }
```

```
    }

    void Sub(Ply &a, Ply &b)
    {
        Ply nB;
        Trm *cur = b.hd->nxt;

        while (cur != b.hd)
        {
            nB.Ins(-cur->cof, cur->exp);
            cur = cur->nxt;
        }

        Add(a, nB);
    }

    void Mul(Ply &a, Ply &b)
    {
        Trm *trmA = a.hd->nxt;

        while (trmA != a.hd)
        {
            Trm *trmB = b.hd->nxt;

            while (trmB != b.hd)
            {
                int c = trmA->cof * trmB->cof;
                int e = trmA->exp + trmB->exp;
                Ins(c, e);
                trmB = trmB->nxt;
            }

            trmA = trmA->nxt;
        }
    }

    float Evl(float x)
    {
        float rslt = 0;
        Trm *cur = hd->nxt;

        while (cur != hd)
        {
            rslt += cur->cof * pow(x, cur->exp);
            cur = cur->nxt;
        }

        return rslt;
    }
```

```
    void Erase(int exp)
    {
        Trm *cur = hd->nxt;
        Trm *prv = hd;

        while (cur != hd)
        {
            if (cur->exp == exp)
            {
                prv->nxt = cur->nxt;
                delete cur;
                cur = prv->nxt;
            }
            else
            {
                prv = cur;
                cur = cur->nxt;
            }
        }
    }
};

int main()
{
    Ply pA, pB, rslt;
    cout << "Enter Poly A:" << endl;
    pA.Rd();
    cout << "Enter Poly B:" << endl;
    pB.Rd();

    cout << "Poly A: ";
    pA.Wr();
    cout << "Poly B: ";
    pB.Wr();

    cout << "Sum of A and B: ";
    rslt.Add(pA, pB);
    rslt.Wr();

    cout << "Diff of A and B: ";
    rslt.Sub(pA, pB);
    rslt.Wr();

    cout << "Product of A and B: ";
    rslt.Mul(pA, pB);
    rslt.Wr();

    float evalPt;
    cout << "Enter value to eval A: ";
```

```
    cin >> evalPt;
    cout << "A(" << evalPt << ") = " << pA.Evl(evalPt) << endl;
```

```
    int exp;
    cout << "Enter exp to erase from A: ";
    cin >> exp;
    pA.Erase(exp);
    cout << "A after erasing term with exp " << exp << ": ";
    pA.Wr();
```

```
    return 0;
}
```