Assignment_04

Name:-Harshal Magar
PRN:-124B1B135
DIV:-B3

Code:-

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

struct node {
   int coeff;
   int power;
   node *next;
};

class poly {
   node *head;
public:
   poly() {
      head = NULL;
   }

   void addRandomTerm() {
      int coeff = rand() % 15 + 1;
      int power = rand() % 10 + 1;

      node *nn = new node{coeff, power, NULL};

      if (!head || power > head->power) {
         nn->next = head;
         head = nn;
      } else {
         node *p = head, *prev = NULL;
         while (p && p->power > power) {
            prev = p;
            p = p->next;
         }

         if (p && p->power == power) {
            p->coeff += coeff;
            delete nn;
         } else {
            nn->next = p;
            prev->next = nn;
```

Assignment_04

Name:-Harshal Magar
PRN:-124B1B135
DIV:-B3

```
        }
    }
}

void display_poly() {
    node *p = head;
    while (p) {
        cout << p->coeff << "X^" << p->power;
        if (p->next && p->next->coeff >= 0)
            cout << " + ";
        p = p->next;
    }
    cout << "\n";
}

poly operator+(poly &p2) {
    poly result;
    node *p1 = this->head;
    node *p2ptr = p2.head;

    while (p1 && p2ptr) {
        if (p1->power == p2ptr->power) {
            result.addTerm(p1->coeff + p2ptr->coeff, p1->power);
            p1 = p1->next;
            p2ptr = p2ptr->next;
        } else if (p1->power > p2ptr->power) {
            result.addTerm(p1->coeff, p1->power);
            p1 = p1->next;
        } else {
            result.addTerm(p2ptr->coeff, p2ptr->power);
            p2ptr = p2ptr->next;
        }
    }

    while (p1) {
        result.addTerm(p1->coeff, p1->power);
        p1 = p1->next;
    }

    while (p2ptr) {
        result.addTerm(p2ptr->coeff, p2ptr->power);
        p2ptr = p2ptr->next;
```

Assignment_04

Name:-Harshal Magar
PRN:-124B1B135
DIV:-B3

```
    }

    return result;
}

poly operator-(poly &p2) {
    poly result;
    node *p1 = this->head;
    node *p2ptr = p2.head;

    while (p1 && p2ptr) {
        if (p1->power == p2ptr->power) {
            result.addTerm(p1->coeff - p2ptr->coeff, p1->power);
            p1 = p1->next;
            p2ptr = p2ptr->next;
        } else if (p1->power > p2ptr->power) {
            result.addTerm(p1->coeff, p1->power);
            p1 = p1->next;
        } else {
            result.addTerm(-p2ptr->coeff, p2ptr->power);
            p2ptr = p2ptr->next;
        }
    }

    while (p1) {
        result.addTerm(p1->coeff, p1->power);
        p1 = p1->next;
    }

    while (p2ptr) {
        result.addTerm(-p2ptr->coeff, p2ptr->power);
        p2ptr = p2ptr->next;
    }

    return result;
}

void addTerm(int coeff, int power) {
    if (coeff == 0) return;

    node *nn = new node{coeff, power, NULL};
```

Assignment_04

Name:-Harshal Magar
PRN:-124B1B135
DIV:-B3

```
        if (!head || power > head->power) {
            nn->next = head;
            head = nn;
        } else {
            node *p = head, *prev = NULL;
            while (p && p->power > power) {
                prev = p;
                p = p->next;
            }

            if (p && p->power == power) {
                p->coeff += coeff;
                delete nn;
            } else {
                nn->next = p;
                prev->next = nn;
            }
        }
    }
};

int main() {
    poly p1, p2;

    for (int i = 0; i < 5; i++) {
        p1.addRandomTerm();
        p2.addRandomTerm();
    }

    cout << "P1: ";
    p1.display_poly();

    cout << "P2: ";
    p2.display_poly();

    poly sum = p1 + p2;
    cout << "P1 + P2: ";
    sum.display_poly();

    poly diff = p1 - p2;
    cout << "P1 - P2: ";
    diff.display_poly();
```

Assignment_04

Name:-Harshal Magar
PRN:-124B1B135
DIV:-B3

```
    return 0;
}
```

Output:-

```
PS C:\Users\Harshal\OneDrive\Desktop\CPP(DSA)> cd "c:\Users\Harshal\OneDrive\Desktop\CPP(DSA)\"
.\Assignment_04 }
P1: 14X^8 + 23X^5 + 11X^3
P2: 4X^9 + 13X^7 + 6X^6 + 2X^2 + 5X^1
P1 + P2: 4X^9 + 14X^8 + 13X^7 + 6X^6 + 23X^5 + 11X^3 + 2X^2 + 5X^1
P1 - P2: -4X^9 + 14X^8-13X^7-6X^6 + 23X^5 + 11X^3-2X^2-5X^1
PS C:\Users\Harshal\OneDrive\Desktop\CPP(DSA)>
```