

POLYNOMIAL ADDITION

Exp. No.:

AIM:

ALGORITHM:



PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

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

void createPoly(int coeff, int pow, struct node **head)
{
    struct node *temp, *trav, *prevTrav = NULL;
    if (*head == NULL)
    {
        temp = (struct node *)malloc(sizeof(struct node));
        temp->coeff = coeff;
        temp->pow = pow;
        temp->next = NULL;
        *head = temp;
    }
    else
    {
        trav = *head;
        prevTrav = NULL;
        while (trav != NULL && trav->pow < pow)
        {
            prevTrav = trav;
            trav = trav->next;
        }
        if (trav != NULL)
        {
            if (trav->pow == pow)
            {
                trav->coeff += coeff;
            }
        }
    }
}
```

```

    }
    else if (prevTrav == NULL)
    {
        temp = (struct node *)malloc(sizeof(struct node));
        temp->coeff = coeff;
        temp->pow = pow;
        temp->next = trav;
        *head = temp;
    }
    else
    {
        temp = (struct node *)malloc(sizeof(struct node));
        temp->coeff = coeff;
        temp->pow = pow;
        temp->next = trav;
        prevTrav->next = temp;
    }
}
else
{
    temp = (struct node *)malloc(sizeof(struct node));
    temp->coeff = coeff;
    temp->pow = pow;
    temp->next = NULL;
    prevTrav->next = temp;
}
}
}

```

```

void polyAdd(struct node *poly1, struct node *poly2, struct node **poly)
{
    struct node *trav1 = poly1;
    struct node *trav2 = poly2;
    while (trav1 != NULL)
    {
        createPoly(trav1->coeff, trav1->pow, poly);
        trav1 = trav1->next;
    }
    while (trav2 != NULL)

```

```

{
    createPoly(trav2->coeff, trav2->pow, poly);
    trav2 = trav2->next;
}
}

```

```

void traverse(struct node *head)
{
    struct node *trav = head;
    printf("Polynomial is: ");
    while (trav != NULL)
    {
        if (trav->pow == 0)
        {
            if (trav->coeff > 0)
            {
                printf("+%d ", trav->coeff);
            }
            else if (trav->coeff < 0)
            {
                printf("%d ", trav->coeff);
            }
        }
        else
        {
            if (trav->coeff > 1)
            {
                printf("+%dx^%d ", trav->coeff, trav->pow);
            }
            else if (trav->coeff < -1)
            {
                printf("%dx^%d ", trav->coeff, trav->pow);
            }
            else if (trav->coeff == -1)
            {
                printf("-x^%d ", trav->pow);
            }
            else
            {

```

```

        printf("+x^%d ", trav->pow);
    }
}
trav = trav->next;
}
}

```

```

int main()
{
    int t1, t2, i, coeff, pow;
    struct node *poly1 = NULL, *poly2 = NULL, *poly = NULL;
    printf("Enter Total Number of Terms in 1st Polynomial: ");
    scanf("%d", &t1);
    for (i = 0; i < t1; i++)
    {
        printf("Enter Coeff and Power :");
        scanf("%d", &coeff);
        scanf("%d", &pow);
        createPoly(coeff, pow, &poly1);
    }
    printf("\nEnter Total Number of Terms in 2nd Polynomial: ");
    scanf("%d", &t2);
    for (i = 0; i < t2; i++)
    {
        printf("Enter Coeff and Power: ");
        scanf("%d", &coeff);
        scanf("%d", &pow);
        createPoly(coeff, pow, &poly2);
    }
    polyAdd(poly1, poly2, &poly);
    traverse(poly);
    getch();
    clrscr();
    return 0;
}

```

OUTPUT:

```
Enter Total Number of Terms in 1st Polynomial: 3
Enter Coeff and Power :2 -8
Enter Coeff and Power :7 5
Enter Coeff and Power :-8 0

Enter Total Number of Terms in 2nd Polynomial: 5
Enter Coeff and Power: 6 3
Enter Coeff and Power: 3 2
Enter Coeff and Power: 7 4
Enter Coeff and Power: 9 -4
Enter Coeff and Power: -4 5
Polynomial is: +2x^-8 +9x^-4 -8 +3x^2 +6x^3 +7x^4 +3x^5 _
```

RESULT: