Name– Shreya Mishra

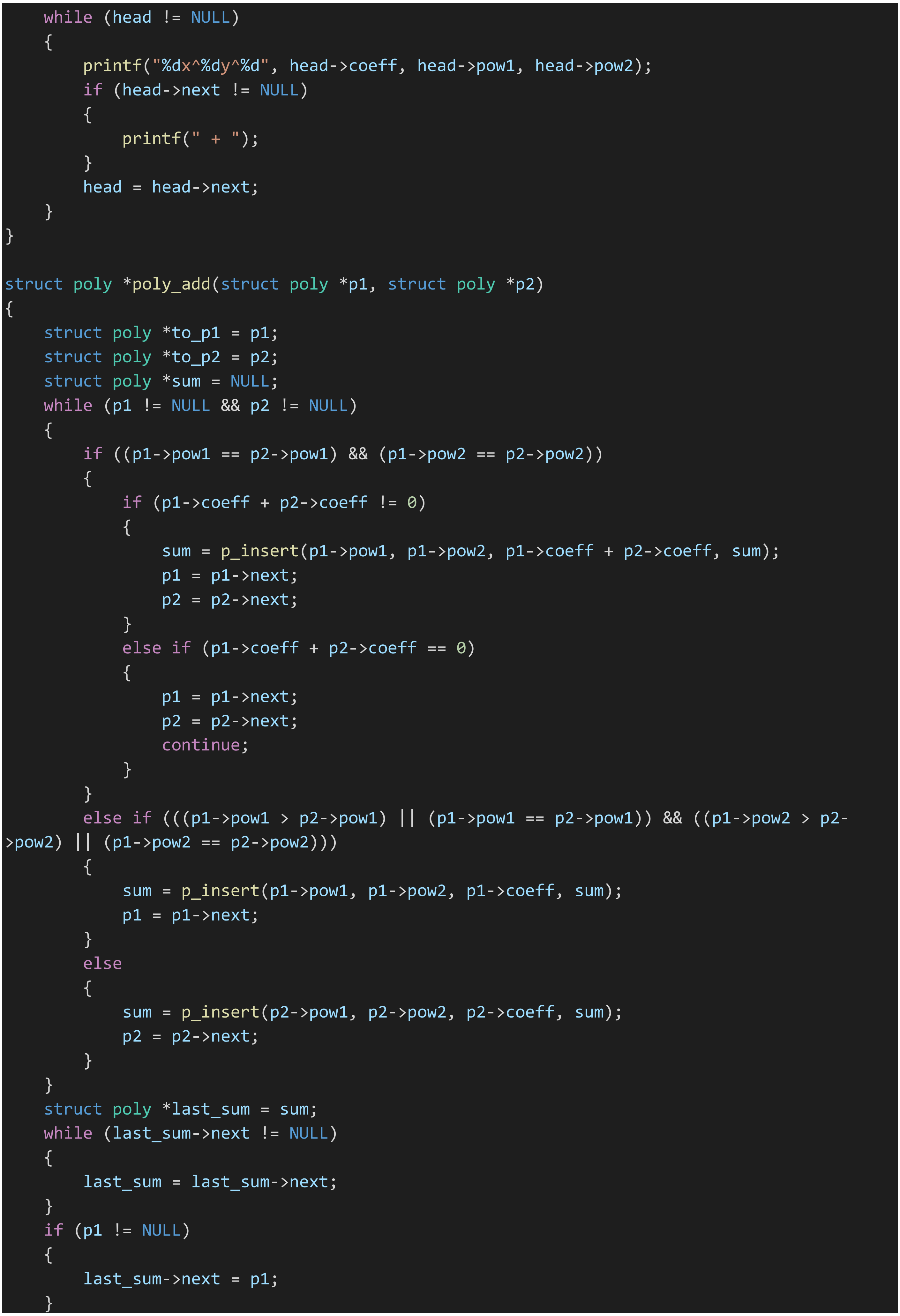
PRN – PES1202102193

SRN – PES1UG21CS574

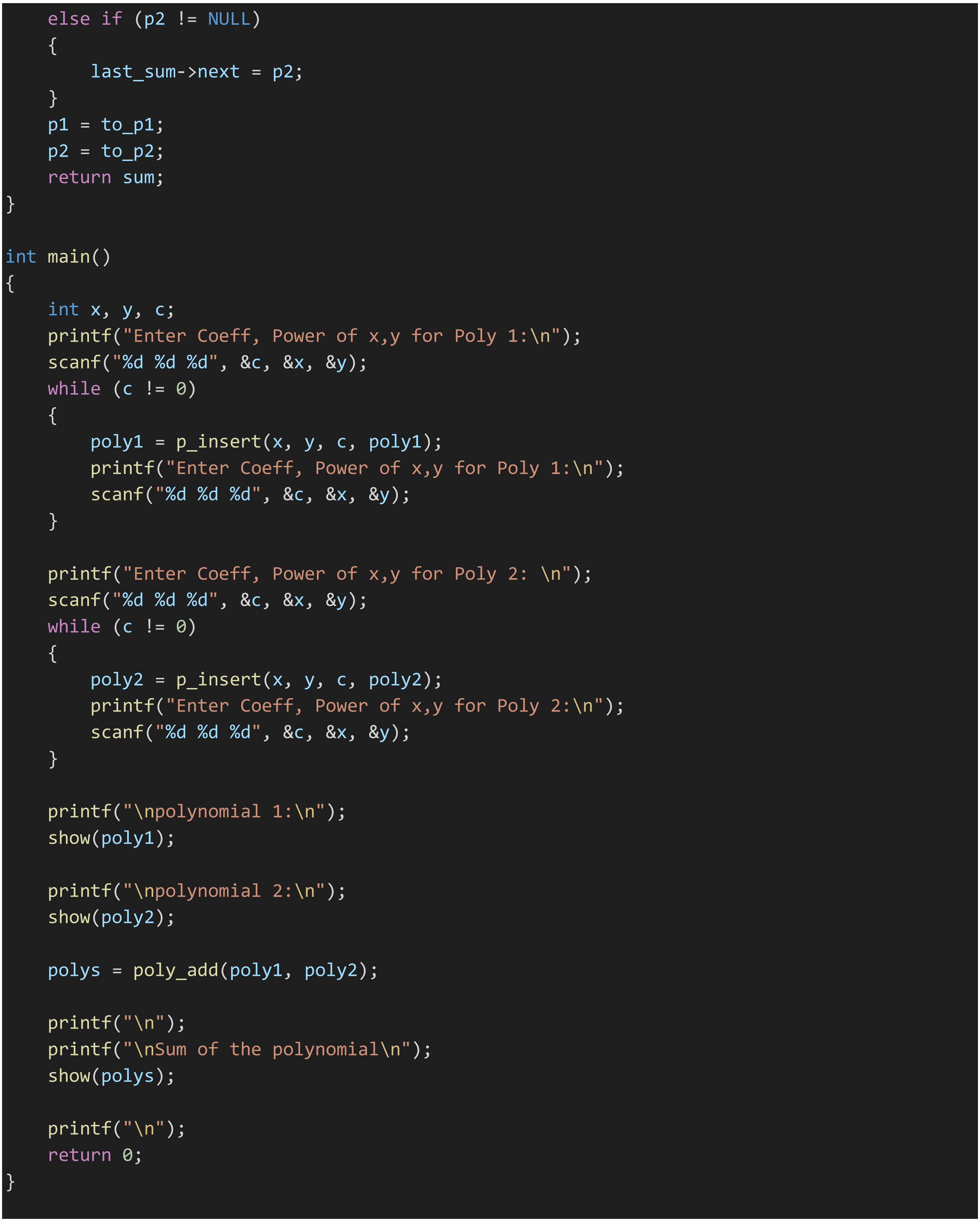
QUES 1

poly.c file

```
#include <stdio.h>
#include <stdlib.h>
struct poly
{
    int pow1;
    int pow2;
    int coeff;
    struct poly *next;
};

struct poly *poly1 = NULL;
struct poly *poly2 = NULL;
struct poly *polys = NULL;

struct poly *p_insert(int x, int y, int c, struct poly *p)
{
    struct poly *temp = (struct poly *)malloc(sizeof(struct poly));
    temp->pow1 = x;
    temp->pow2 = y;
    temp->coeff = c;
    temp->next = NULL;
    if (p == NULL)
    {
        p = temp;
        return p;
    }
    struct poly *ptr = p;
    while (ptr->next != NULL)
    {
        ptr = ptr->next;
    }
    ptr->next = temp;
    return p;
}

void show(struct poly *p)
{
    if (p == NULL)
    {
        printf("Polynomial is Empty!\n");
        return;
    }
    struct poly *head = p;
```

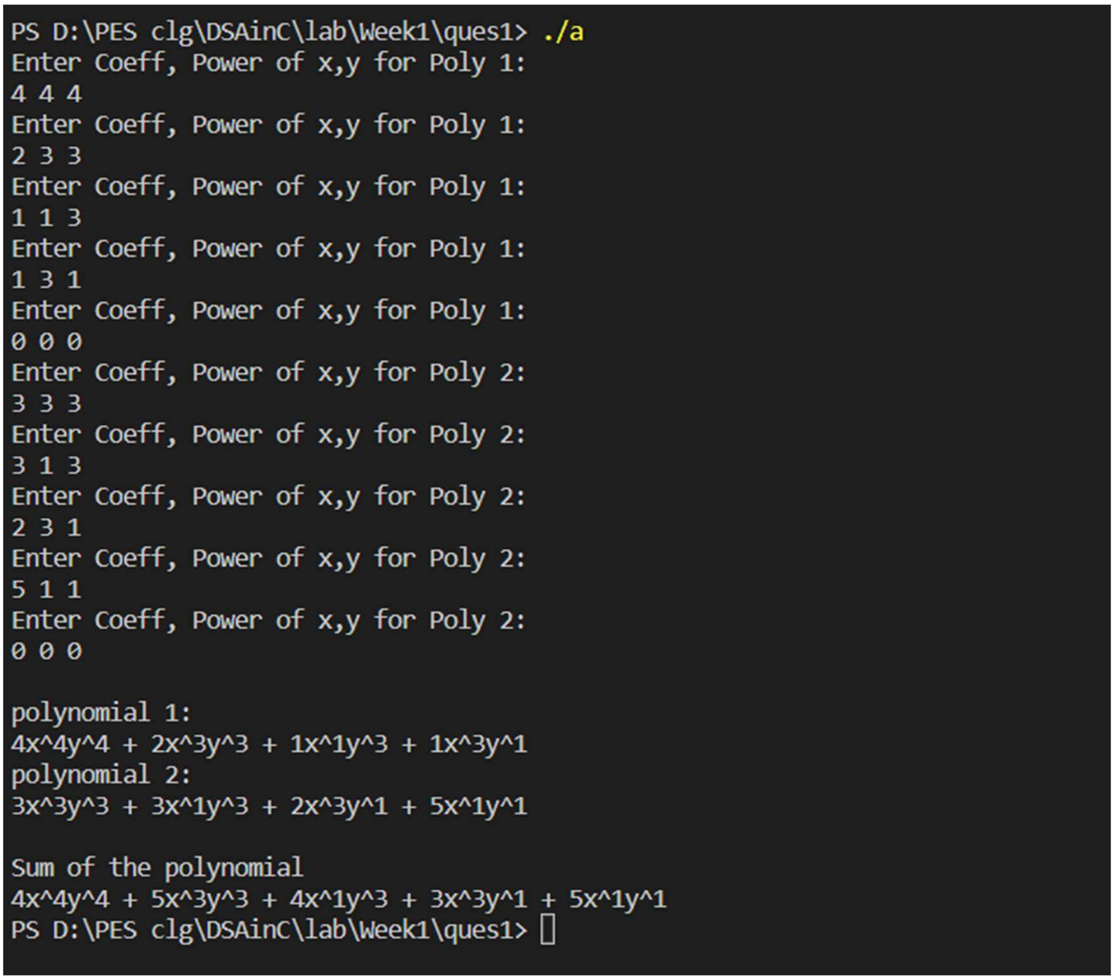


```
    while (head != NULL)
    {
        printf("%dx^%dy^%d", head->coeff, head->pow1, head->pow2);
        if (head->next != NULL)
        {
            printf(" + ");
        }
        head = head->next;
    }
}

struct poly *poly_add(struct poly *p1, struct poly *p2)
{
    struct poly *to_p1 = p1;
    struct poly *to_p2 = p2;
    struct poly *sum = NULL;
    while (p1 != NULL && p2 != NULL)
    {
        if ((p1->pow1 == p2->pow1) && (p1->pow2 == p2->pow2))
        {
            if (p1->coeff + p2->coeff != 0)
            {
                sum = p_insert(p1->pow1, p1->pow2, p1->coeff + p2->coeff, sum);
                p1 = p1->next;
                p2 = p2->next;
            }
            else if (p1->coeff + p2->coeff == 0)
            {
                p1 = p1->next;
                p2 = p2->next;
                continue;
            }
        }
        else if (((p1->pow1 > p2->pow1) || (p1->pow1 == p2->pow1)) && ((p1->pow2 > p2-
>pow2) || (p1->pow2 == p2->pow2)))
        {
            sum = p_insert(p1->pow1, p1->pow2, p1->coeff, sum);
            p1 = p1->next;
        }
        else
        {
            sum = p_insert(p2->pow1, p2->pow2, p2->coeff, sum);
            p2 = p2->next;
        }
    }
    struct poly *last_sum = sum;
    while (last_sum->next != NULL)
    {
        last_sum = last_sum->next;
    }
    if (p1 != NULL)
    {
        last_sum->next = p1;
    }
```



```c
    else if (p2 != NULL)
    {
        last_sum->next = p2;
    }
    p1 = to_p1;
    p2 = to_p2;
    return sum;
}

int main()
{
    int x, y, c;
    printf("Enter Coeff, Power of x,y for Poly 1:\n");
    scanf("%d %d %d", &c, &x, &y);
    while (c != 0)
    {
        poly1 = p_insert(x, y, c, poly1);
        printf("Enter Coeff, Power of x,y for Poly 1:\n");
        scanf("%d %d %d", &c, &x, &y);
    }

    printf("Enter Coeff, Power of x,y for Poly 2: \n");
    scanf("%d %d %d", &c, &x, &y);
    while (c != 0)
    {
        poly2 = p_insert(x, y, c, poly2);
        printf("Enter Coeff, Power of x,y for Poly 2:\n");
        scanf("%d %d %d", &c, &x, &y);
    }

    printf("\npolynomial 1:\n");
    show(poly1);

    printf("\npolynomial 2:\n");
    show(poly2);

    polys = poly_add(poly1, poly2);

    printf("\n");
    printf("\nSum of the polynomial\n");
    show(polys);

    printf("\n");
    return 0;
}
```

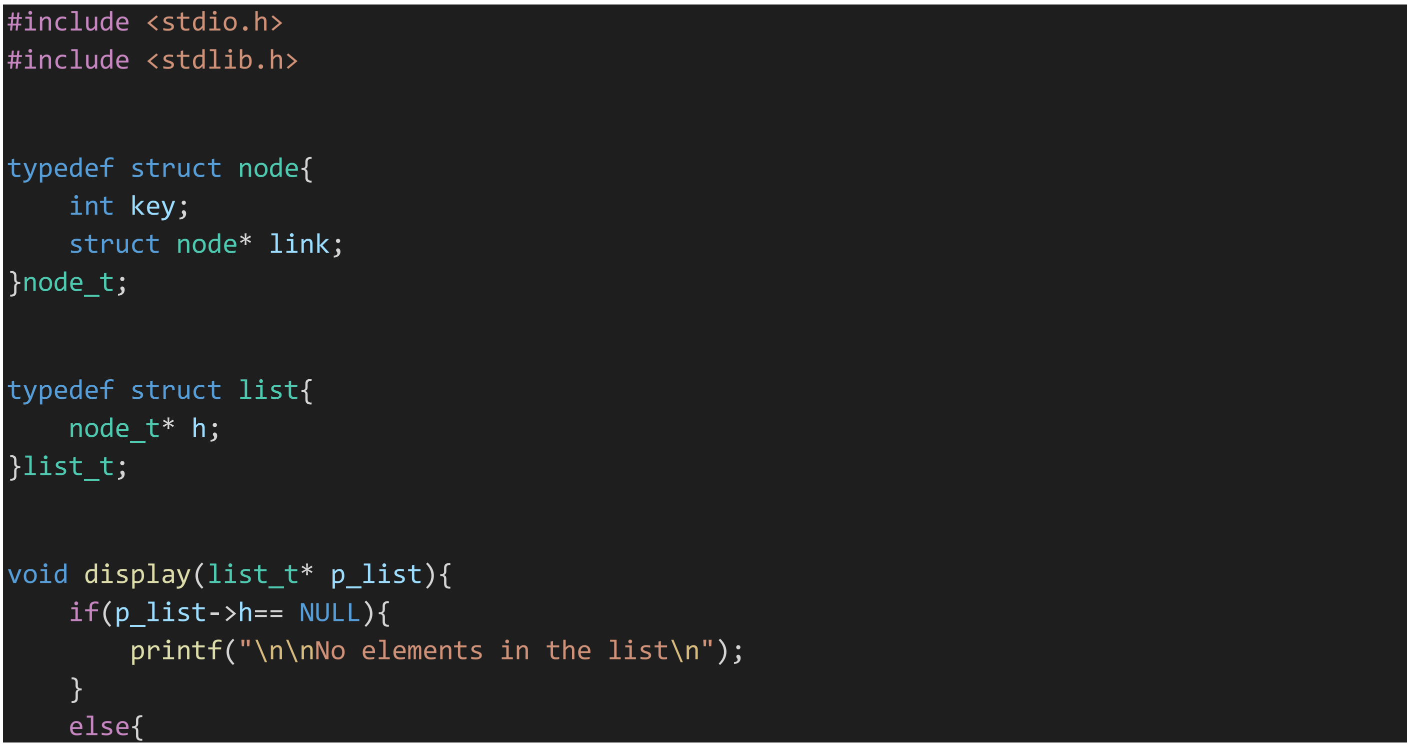
Output



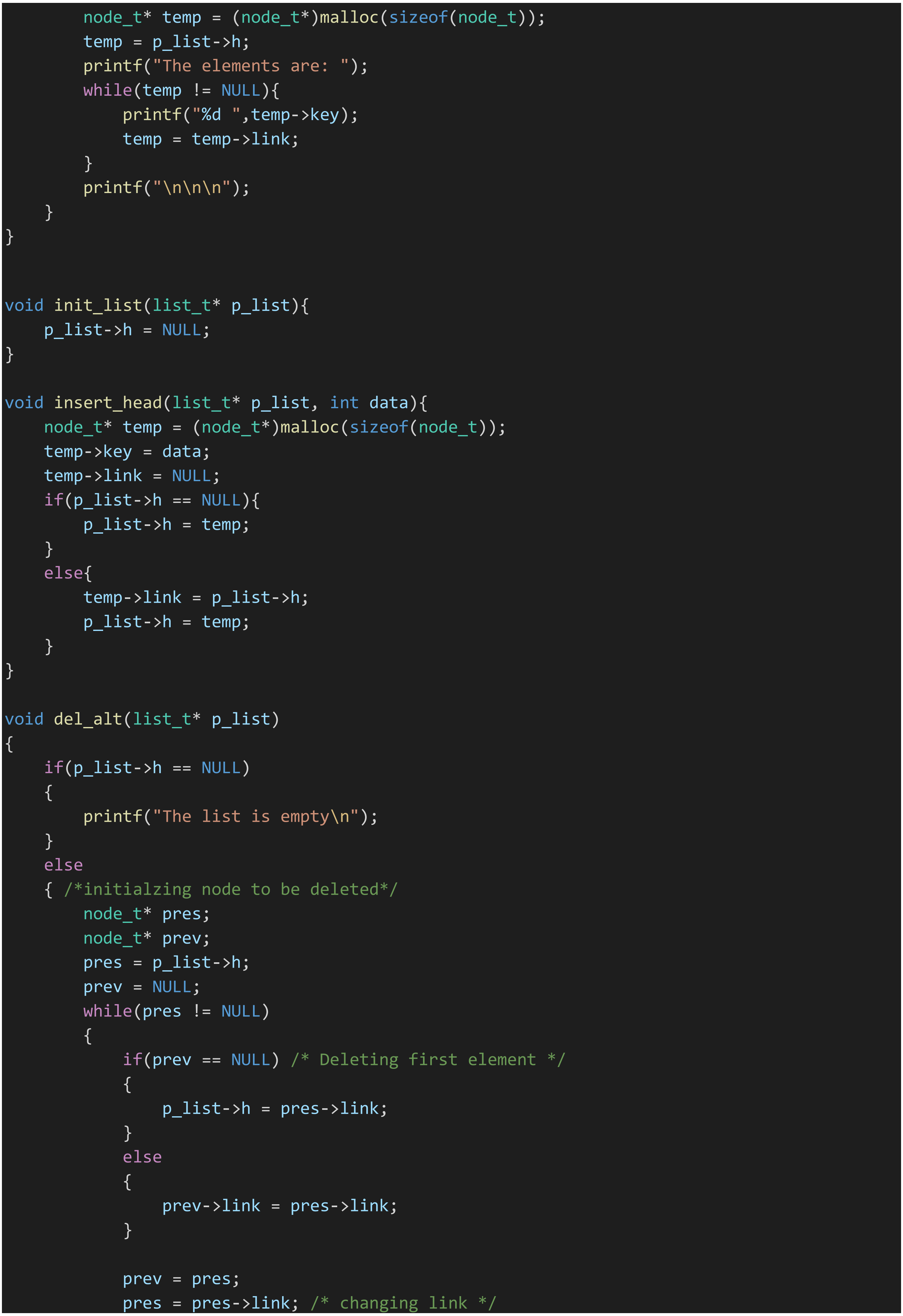
```
PS D:\PES clg\DSAinC\lab\Week1\ques1> ./a
Enter Coeff, Power of x,y for Poly 1:
4 4 4
Enter Coeff, Power of x,y for Poly 1:
2 3 3
Enter Coeff, Power of x,y for Poly 1:
1 1 3
Enter Coeff, Power of x,y for Poly 1:
1 3 1
Enter Coeff, Power of x,y for Poly 1:
0 0 0
Enter Coeff, Power of x,y for Poly 2:
3 3 3
Enter Coeff, Power of x,y for Poly 2:
3 1 3
Enter Coeff, Power of x,y for Poly 2:
2 3 1
Enter Coeff, Power of x,y for Poly 2:
5 1 1
Enter Coeff, Power of x,y for Poly 2:
0 0 0

polynomial 1:
4x^4y^4 + 2x^3y^3 + 1x^1y^3 + 1x^3y^1
polynomial 2:
3x^3y^3 + 3x^1y^3 + 2x^3y^1 + 5x^1y^1

Sum of the polynomial
4x^4y^4 + 5x^3y^3 + 4x^1y^3 + 3x^3y^1 + 5x^1y^1
PS D:\PES clg\DSAinC\lab\Week1\ques1> 
```

QUES-2

deleteAlt.c

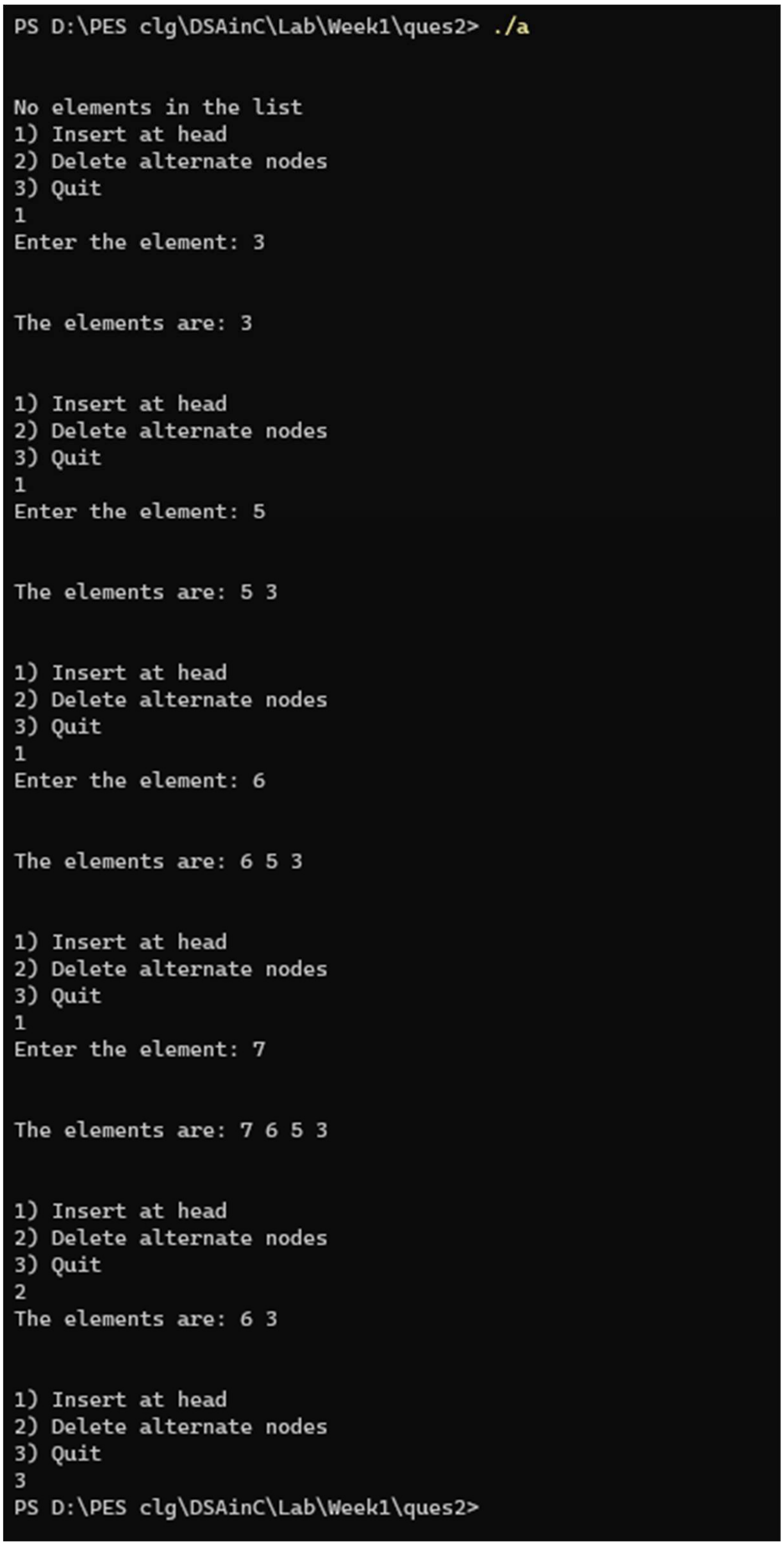


```c
#include <stdio.h>
#include <stdlib.h>


typedef struct node{
    int key;
    struct node* link;
}node_t;


typedef struct list{
    node_t* h;
}list_t;


void display(list_t* p_list){
    if(p_list->h== NULL){
        printf("\n\nNo elements in the list\n");
    }
    else{
```



```
        node_t* temp = (node_t*)malloc(sizeof(node_t));
        temp = p_list->h;
        printf("The elements are: ");
        while(temp != NULL){
            printf("%d ",temp->key);
            temp = temp->link;
        }
        printf("\n\n\n");
    }
}


void init_list(list_t* p_list){
    p_list->h = NULL;
}

void insert_head(list_t* p_list, int data){
    node_t* temp = (node_t*)malloc(sizeof(node_t));
    temp->key = data;
    temp->link = NULL;
    if(p_list->h == NULL){
        p_list->h = temp;
    }
    else{
        temp->link = p_list->h;
        p_list->h = temp;
    }
}

void del_alt(list_t* p_list)
{
    if(p_list->h == NULL)
    {
        printf("The list is empty\n");
    }
    else
    { /*initialzing node to be deleted*/
        node_t* pres;
        node_t* prev;
        pres = p_list->h;
        prev = NULL;
        while(pres != NULL)
        {
            if(prev == NULL) /* Deleting first element */
            {
                p_list->h = pres->link;
            }
            else
            {
                prev->link = pres->link;
            }

            prev = pres;
            pres = pres->link; /* changing link */
        }
    }
}


int main()
{
    list_t l;
    init_list(&l);
    int choice, ele;
    while(1)
    {
        display(&l);
        printf("1) Insert at head\n");
        printf("2) Delete alternate nodes\n");
        printf("3) Quit\n");
        scanf("%d",&choice);
        switch(choice){
            case 1:
                printf("Enter the element: ");
                scanf("%d",&ele);
                printf("\n\n");
                insert_head(&l,ele);
                break;
            case 2:
                del_alt(&l);
                break;
            case 3:
                exit(0);
                break;
            default:
                printf("Invalid Choice \n");
                break;

        }
    }
}
```

OUTPUT



```
PS D:\PES clg\DSAinC\Lab\Week1\ques2> ./a


No elements in the list
1) Insert at head
2) Delete alternate nodes
3) Quit
1
Enter the element: 3


The elements are: 3


1) Insert at head
2) Delete alternate nodes
3) Quit
1
Enter the element: 5


The elements are: 5 3


1) Insert at head
2) Delete alternate nodes
3) Quit
1
Enter the element: 6


The elements are: 6 5 3


1) Insert at head
2) Delete alternate nodes
3) Quit
1
Enter the element: 7


The elements are: 7 6 5 3


1) Insert at head
2) Delete alternate nodes
3) Quit
2
The elements are: 6 3


1) Insert at head
2) Delete alternate nodes
3) Quit
3
PS D:\PES clg\DSAinC\Lab\Week1\ques2>
```