

Name: Gaurav singh

Roll No: 70 Semester: 3 Sem.

Subject: Data structures and Algorithms.

Practical: 5 Implement a program to add two polynomials represented as linked lists A and B to get resultant polynomial represented as linked lists C.

**Code:**

```
#include <bits/stdc++.h>

using namespace std;

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

// Function to create new node
void create_node(int x, int y, struct Node** temp)
{
    struct Node *r, *z;
    z = *temp;
    if (z == NULL) {
        r = (struct Node*)malloc(sizeof(struct Node));
        r->coeff = x;
        r->pow = y;
        *temp = r;
        r->next = (struct Node*)malloc(sizeof(struct Node));
        r = r->next;
        r->next = NULL;
    }
```

```

else {
    r->coeff = x;
    r->pow = y;
    r->next = (struct Node*)malloc(sizeof(struct Node));
    r = r->next;
    r->next = NULL;
}
}

```

```

void polyadd(struct Node* poly1, struct Node* poly2,
             struct Node* poly)
{
    while (poly1->next && poly2->next) {

        if (poly1->pow > poly2->pow) {
            poly->pow = poly1->pow;
            poly->coeff = poly1->coeff;
            poly1 = poly1->next;
        }

        // If power of 2nd polynomial is greater then 1st,
        // then store 2nd as it is and move its pointer
        else if (poly1->pow < poly2->pow) {
            poly->pow = poly2->pow;
            poly->coeff = poly2->coeff;
            poly2 = poly2->next;
        }
    }
}

```

```
else {  
    poly->pow = poly1->pow;  
    poly->coeff = poly1->coeff + poly2->coeff;  
    poly1 = poly1->next;  
    poly2 = poly2->next;  
}
```

```
poly->next  
    = (struct Node*)malloc(sizeof(struct Node));  
poly = poly->next;  
poly->next = NULL;  
}
```

```
while (poly1->next || poly2->next) {
```

```
    if (poly1->next) {  
        poly->pow = poly1->pow;  
        poly->coeff = poly1->coeff;  
        poly1 = poly1->next;  
    }
```

```
    if (poly2->next) {  
        poly->pow = poly2->pow;  
        poly->coeff = poly2->coeff;  
        poly2 = poly2->next;  
    }
```

```
poly->next  
    = (struct Node*)malloc(sizeof(struct Node));  
poly = poly->next;
```

```
    poly->next = NULL;
}
}
```

```
void show(struct Node* node)
{
    while (node->next != NULL) {
        printf("%dx^%d", node->coeff, node->pow);
        node = node->next;
        if (node->coeff >= 0) {
            if (node->next != NULL)
                printf("+");
        }
    }
}
```

```
int main()
{
    struct Node *poly1 = NULL, *poly2 = NULL, *poly = NULL;

    create_node(5, 2, &poly1);
    create_node(4, 1, &poly1);
    create_node(2, 0, &poly1);
```

```
create_node(-5, 1, &poly2);
```

```
create_node(-5, 0, &poly2);
```

```
printf("1st Number: ");
```

```
show(poly1);
```

```
printf("\n2nd Number: ");
```

```
show(poly2);
```

```
poly = (struct Node*)malloc(sizeof(struct Node));
```

```
polyadd(poly1, poly2, poly);
```

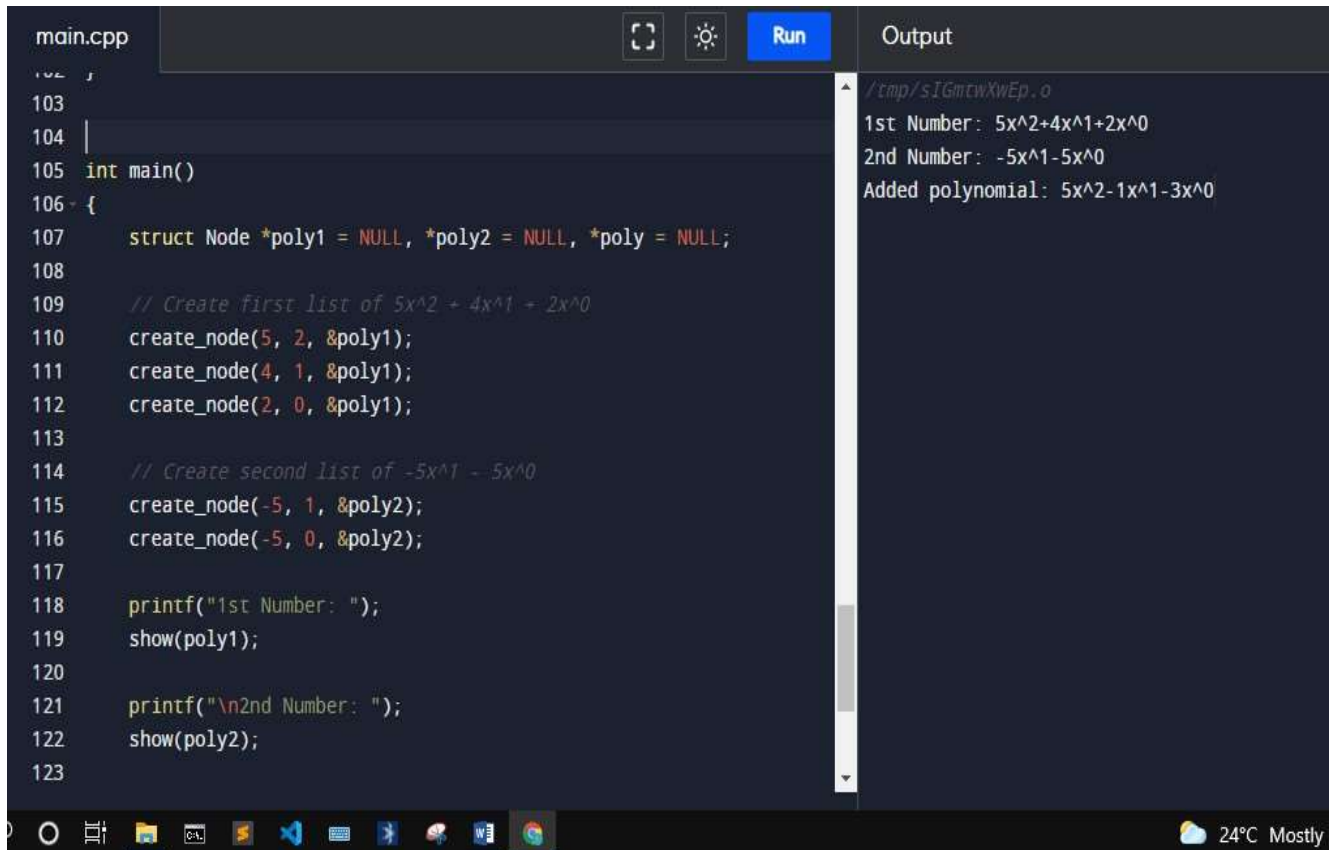
```
printf("\nAdded polynomial: ");
```

```
show(poly);
```

```
return 0;
```

```
}
```

## Output:



The image shows a C++ IDE with a dark theme. The editor window displays a file named `main.cpp` with line numbers 102 to 123. The code defines a linked list structure to represent polynomials. It creates two polynomials: the first is  $5x^2 + 4x^1 + 2x^0$  and the second is  $-5x^1 - 5x^0$ . The program then prints each polynomial and the result of their addition, which is  $5x^2 - 1x^1 - 3x^0$ . The output window on the right shows the execution results. The Windows taskbar at the bottom includes icons for various applications and a system tray showing the temperature as 24°C and weather as Mostly.

```
102 ,
103
104 |
105 int main()
106 {
107     struct Node *poly1 = NULL, *poly2 = NULL, *poly = NULL;
108
109     // Create first list of 5x^2 + 4x^1 + 2x^0
110     create_node(5, 2, &poly1);
111     create_node(4, 1, &poly1);
112     create_node(2, 0, &poly1);
113
114     // Create second list of -5x^1 - 5x^0
115     create_node(-5, 1, &poly2);
116     create_node(-5, 0, &poly2);
117
118     printf("1st Number: ");
119     show(poly1);
120
121     printf("\n2nd Number: ");
122     show(poly2);
123 }
```

Output

/tmp/SIGmcwXwEp.o  
1st Number: 5x^2+4x^1+2x^0  
2nd Number: -5x^1-5x^0  
Added polynomial: 5x^2-1x^1-3x^0

24°C Mostly