

PRACTICAL 5

Name: Mithlesh Yeole

Batch: B3-B3

Roll no. : 59

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

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

```
struct node{
```

```
    float co;
```

```
    int expo;
```

```
    struct node*link;
```

```
};
```

```
struct node* insert(struct node* head, float co, int expo) {
```

```
    struct node* temp = (struct node*)malloc(sizeof(struct node));
```

```
    temp->co = co;
```

```
    temp->expo = expo;
```

```
    temp->link = NULL;
```

```
    if (head == NULL || head->expo < expo) {
```

```
        temp->link = head;
```

```
        return temp;
```

```
    }
```

```
    struct node* temp1 = head;
```

```
    while (temp1->link != NULL && temp1->link->expo > expo) {
```

```
        temp1 = temp1->link;
```

```
}
```

```
if (temp1->link != NULL && temp1->link->expo == expo) {
```

```
    temp1->link->co += co;
```

```
    free(temp);
```

```
} else {
```

```
    temp->link = temp1->link;
```

```
    temp1->link = temp;
```

```
}
```

```
return head;
```

```
};
```

```
struct node* create() {
```

```
    struct node* head = NULL;
```

```
    int n, expo;
```

```
    float co;
```

```
    printf("Enter the number of terms: ");
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++) {
```

```
        printf("Enter coefficient and exponent: ");
```

```
        scanf("%f %d", &co, &expo);
```

```
        head = insert(head, co, expo);
```

```
    }
```

```
    return head;
```

```
}
```

```

struct node* add(struct node* head, struct node* head1){
    struct node* temp=head1;
    struct node* temp1=head1;
    struct node* head2=NULL;
    while (temp != NULL && temp1 != NULL) {
        if (temp1->expo == temp1->expo) {
            head2 = insert(head2, temp1->co, temp1->expo);
            temp1 = temp1->link;
        } else if (temp->expo > temp1->expo) {
            head2 = insert(head2, temp->co, temp->expo);
            temp = temp->link;
        } else if (temp1->expo>temp->expo){
            head2 = insert(head2, temp1->co + temp1->co, temp1->expo);
            temp1 = temp1->link;
            temp1 = temp1->link;
        }
    }
    while(temp!=NULL){
        head2=insert(head2,temp->co,temp->expo);
        temp=temp->link;
    }
    return head2;
}

void print(struct node* head) {
    if (head == NULL) {
        printf("0\n");
        return;
    }

    struct node* temp = head;
    while (temp != NULL) {

```

```

        printf("%.1fx^%d", temp->co, temp->expo);
        if (temp->link != NULL)
            printf(" + ");
        temp = temp->link;
    }
    printf("\n");
}

```

```

void freeList(struct node* head) {
    struct node* temp;
    while (head != NULL) {
        temp = head;
        head = head->link;
        free(temp);
    }
}

```

```

int main() {
    struct node* poly1 = NULL;
    struct node* poly2 = NULL;
    struct node* result = NULL;

    printf("Enter Polynomial 1:\n");
    poly1 = create();

    printf("Enter Polynomial 2:\n");
    poly2 = create();

    printf("Polynomial 1: ");

```

```
print(poly1);
```

```
printf("Polynomial 2: ");
```

```
print(poly2);
```

```
result = add(poly1, poly2);
```

```
printf("Sum: ");
```

```
print(result);
```

```
freeList(poly1);
```

```
freeList(poly2);
```

```
freeList(result);
```

```
return 0;
```

```
}
```

Output:

The screenshot displays the OnlineGDB website's IDE. On the left is a blue sidebar menu with links: "OnlineGDB" (with a lightning bolt icon), "online compiler and debugger for c/c++", "code.compile.run.debug.share.", "IDE", "My Projects", "Classroom new", "Learn Programming", "Programming Questions", "Sign Up", and "Login". The main area has a top toolbar with icons for Run, Debug, Stop, Share, Save, Beautify, and Update. Below the toolbar, the editor shows a file named "main.c" with the following C code:

```
111 struct node* result = NULL;
112
113 printf("Enter Polynomial 1:\n");
114 poly1 = create();
115
116 //printf("Enter Polynomial 2:\n");
```

The console output at the bottom shows the program's execution:

```
Enter Polynomial 1:
Enter the number of terms: 3
Enter coefficient and exponent: 2 2
Enter coefficient and exponent: 3 1
Enter coefficient and exponent: 4 0
Enter Polynomial 2:
Enter the number of terms: 3
Enter coefficient and exponent: 3 5
Enter coefficient and exponent: 2 2
Enter coefficient and exponent: 3 0
Polynomial 1: 2.0x^2 + 3.0x^1 + 4.0x^0
Polynomial 2: 3.0x^5 + 2.0x^2 + 3.0x^0
Sum: 3.0x^5 + 3.0x^5 + 4.0x^2 + 6.0x^0

...Program finished with exit code 0
Press ENTER to exit console.
```