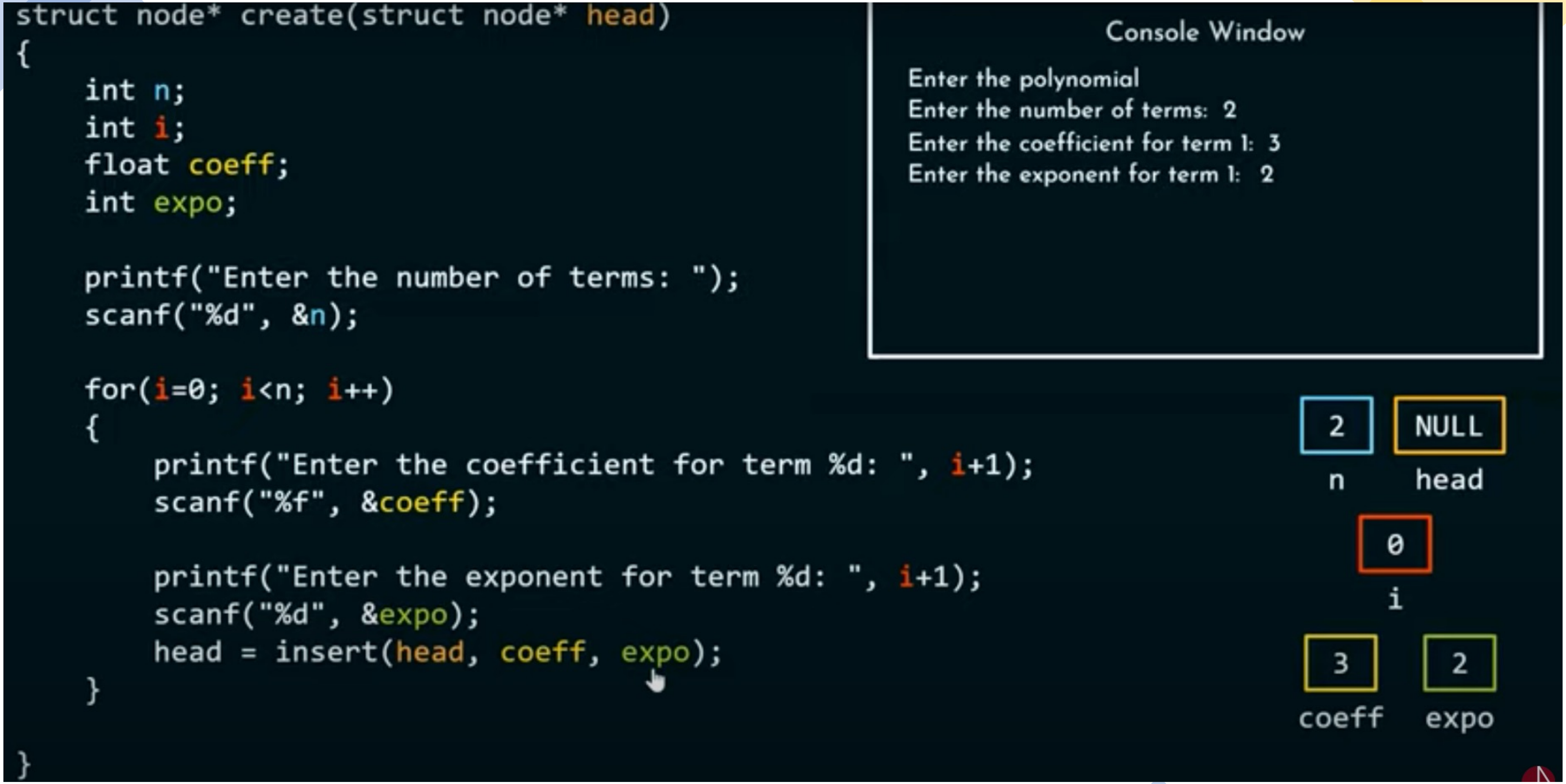
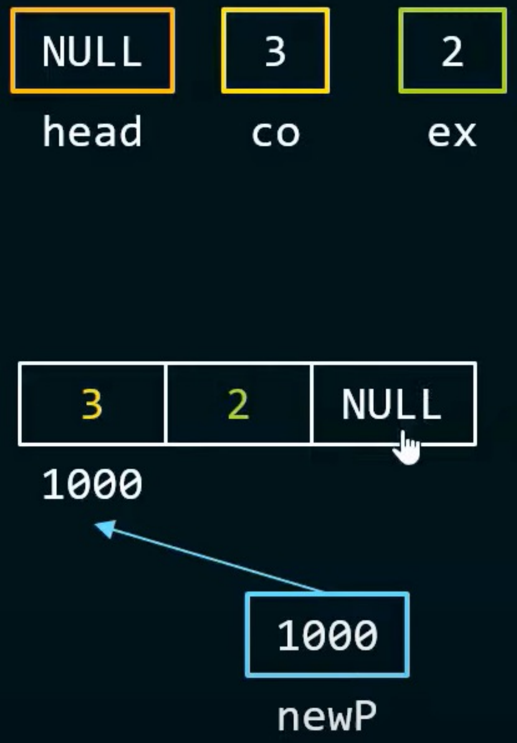


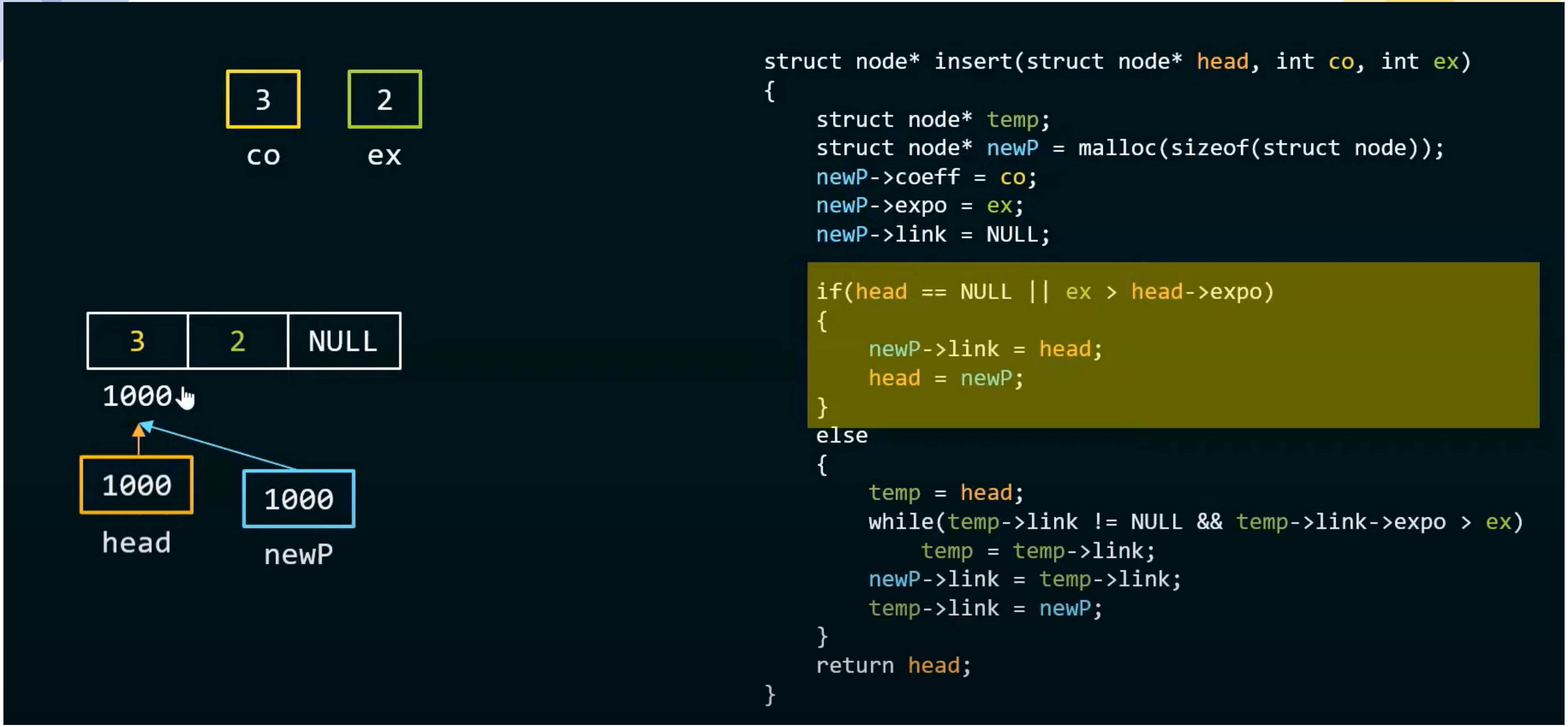
```
struct node* create(struct node* head)
{
    int n;
    int i;
    float coeff;
    int expo;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    for(i=0; i<n; i++)
    {
        printf("Enter the coefficient for term %d: ", i+1);
        scanf("%f", &coeff);

        printf("Enter the exponent for term %d: ", i+1);
        scanf("%d", &expo);
        head = insert(head, coeff, expo);
    }

}
```

Console Window

```
Enter the polynomial
Enter the number of terms:  2
Enter the coefficient for term 1:  3
Enter the exponent for term 1:  2
```



```
struct node* insert(struct node* head, int co, int ex)
{
    struct node* temp;
    struct node* newP = malloc(sizeof(struct node));
    newP->coeff = co;
    newP->expo = ex;
    newP->link = NULL;

    if(head == NULL || ex > head->expo)
    {
        newP->link = head;
        head = newP;
    }
    else
    {
        temp = head;
        while(temp->link != NULL && temp->link->expo > ex)
            temp = temp->link;
        newP->link = temp->link;
        temp->link = newP;
    }
    return head;
}
```

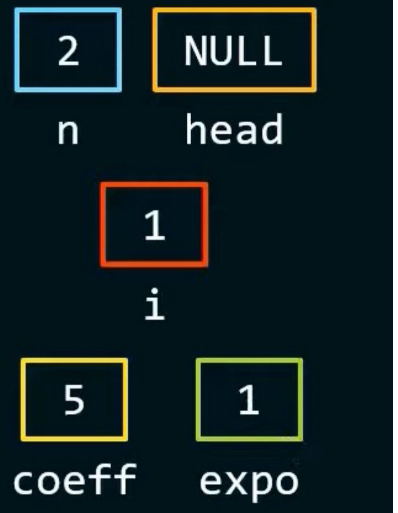


```
struct node* insert(struct node* head, int co, int ex)
{
    struct node* temp;
    struct node* newP = malloc(sizeof(struct node));
    newP->coeff = co;
    newP->expo = ex;
    newP->link = NULL;

    if(head == NULL || ex > head->expo)
    {
        newP->link = head;
        head = newP;
    }
    else
    {
        temp = head;
        while(temp->link != NULL && temp->link->expo > ex)
            temp = temp->link;
        newP->link = temp->link;
        temp->link = newP;
    }
    return head;
}
```

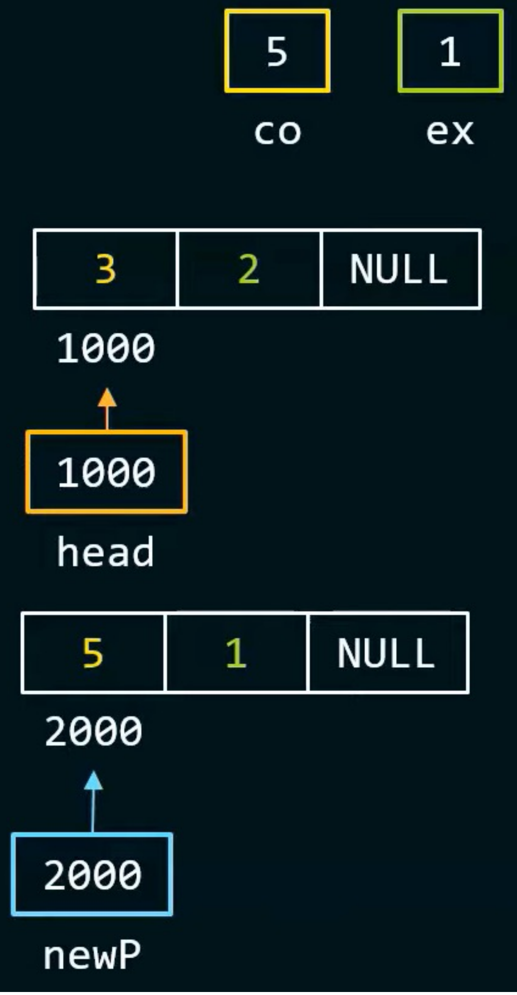
```c
struct node* create(struct node* head)
{
    int n;
    int i;
    float coeff;
    int expo;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    for(i=0; i<n; i++)
    {
        printf("Enter the coefficient for term %d: ", i+1);
        scanf("%f", &coeff);

        printf("Enter the exponent for term %d: ", i+1);
        scanf("%d", &expo);
        head = insert(head, coeff, expo);
    }

}
```

Console Window

```
Enter the polynomial
Enter the number of terms:  2
Enter the coefficient for term 1:  3
Enter the exponent for term 1:  2
Enter the coefficient for term 2:  5
Enter the exponent for term 2:  1
```

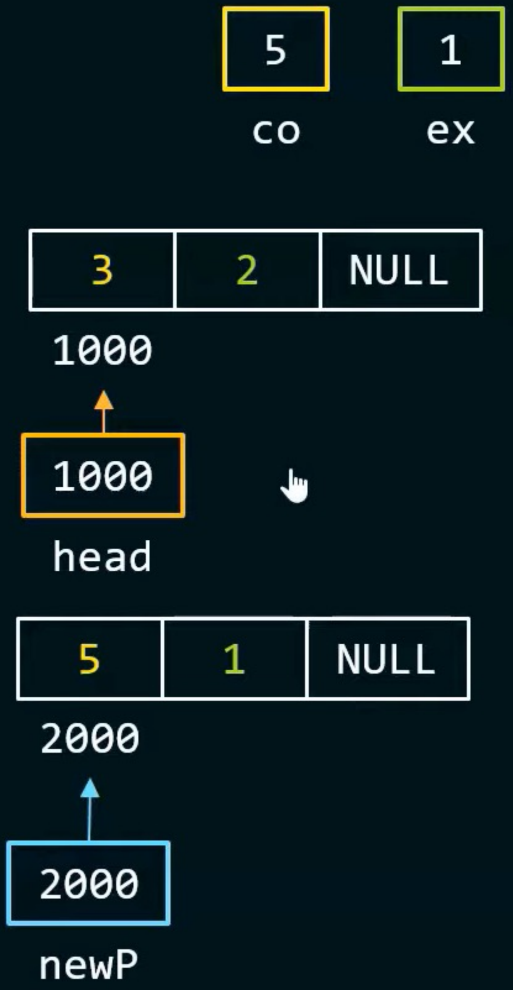


New Insert Function



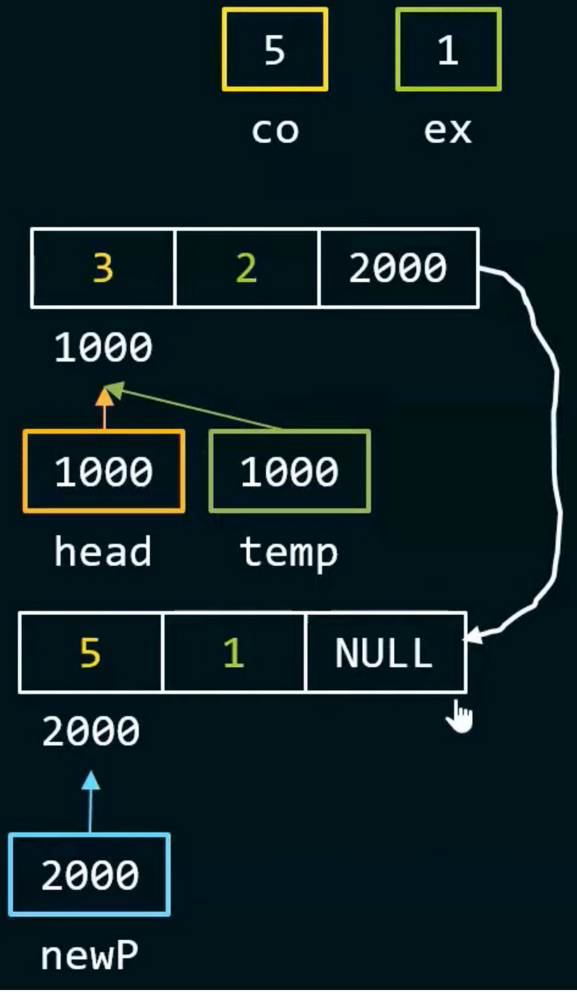
```
struct node* insert(struct node* head, int co, int ex)
{
    struct node* temp;
    struct node* newP = malloc(sizeof(struct node));
    newP->coeff = co;
    newP->expo = ex;
    newP->link = NULL;

    if(head == NULL || ex > head->expo)
    {
        newP->link = head;
        head = newP;
    }
    else
    {
        temp = head;
        while(temp->link != NULL && temp->link->expo > ex)
            temp = temp->link;
        newP->link = temp->link;
        temp->link = newP;
    }
    return head;
}
```

New Insert Function



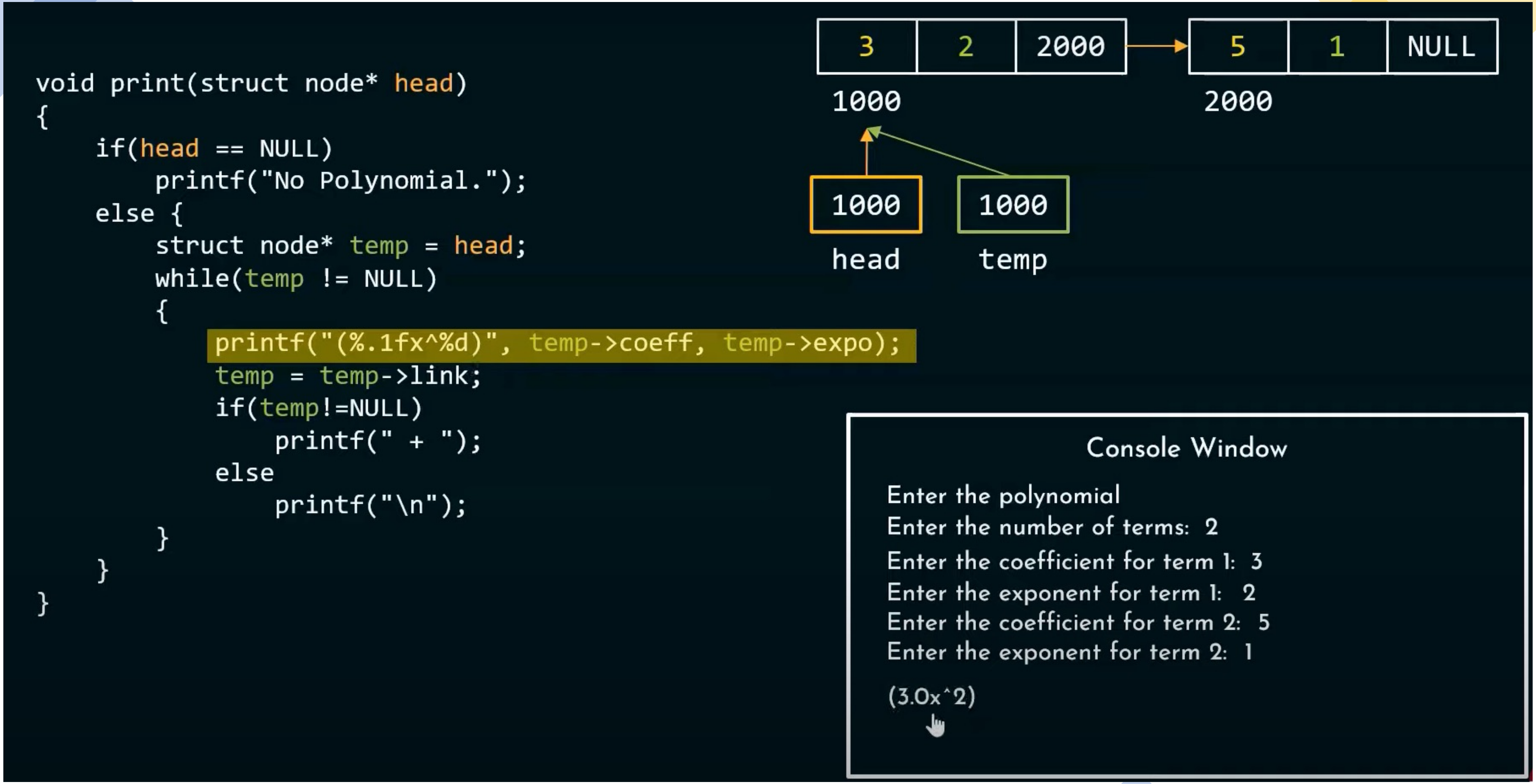
```
struct node* insert(struct node* head, int co, int ex)
{
    struct node* temp;
    struct node* newP = malloc(sizeof(struct node));
    newP->coeff = co;
    newP->expo = ex;
    newP->link = NULL;

    if(head == NULL || ex > head->expo)
    {
        newP->link = head;
        head = newP;
    }
    else
    {
        temp = head;
        while(temp->link != NULL && temp->link->expo > ex)
            temp = temp->link;
        newP->link = temp->link;
        temp->link = newP;
    }
    return head;
}
```

New Insert Function



```
struct node* insert(struct node* head, int co, int ex)
{
    struct node* temp;
    struct node* newP = malloc(sizeof(struct node));
    newP->coeff = co;
    newP->expo = ex;
    newP->link = NULL;

    if(head == NULL || ex > head->expo)
    {
        newP->link = head;
        head = newP;
    }
    else
    {
        temp = head;
        while(temp->link != NULL && temp->link->expo > ex)
            temp = temp->link;
        newP->link = temp->link;
        temp->link = newP;
    }
    return head;
}
```

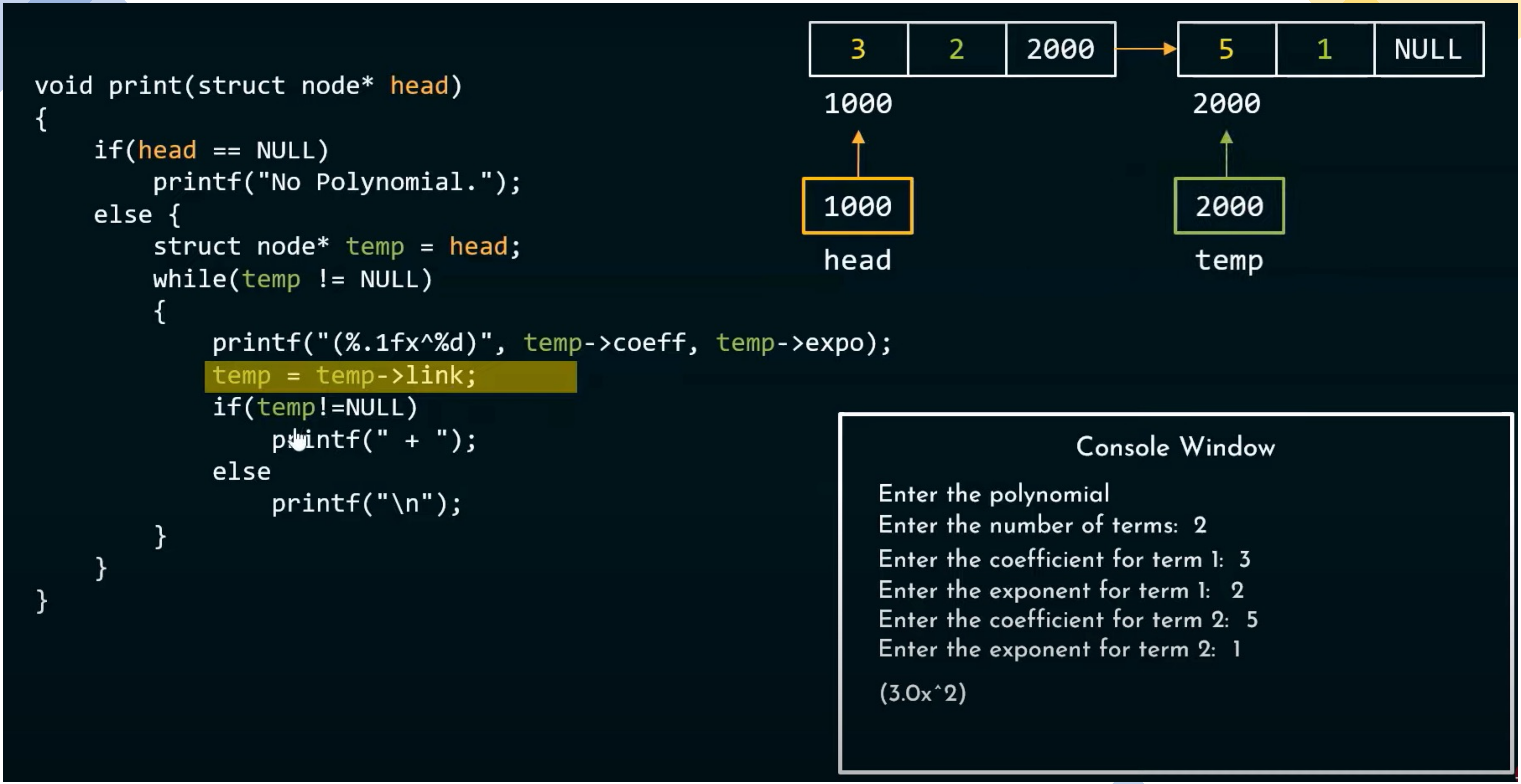
Display Polynomial



```c
void print(struct node* head)
{
    if(head == NULL)
        printf("No Polynomial.");
    else {
        struct node* temp = head;
        while(temp != NULL)
        {
            printf("(%.1fx^%d)", temp->coeff, temp->expo);
            temp = temp->link;
            if(temp!=NULL)
                printf(" + ");
            else
                printf("\n");
        }
    }
}
```

Console Window

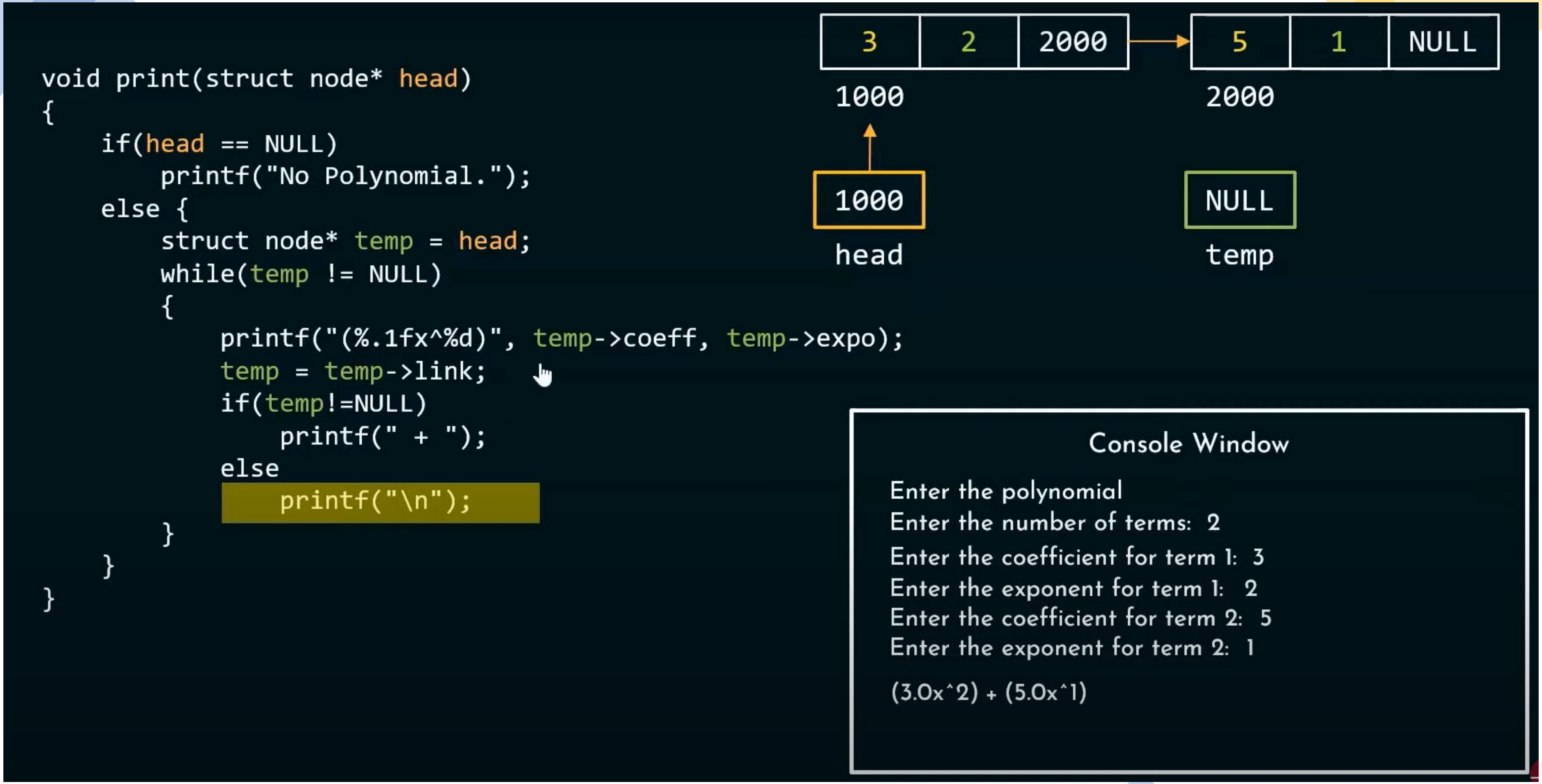
```
Enter the polynomial
Enter the number of terms:  2
Enter the coefficient for term 1:  3
Enter the exponent for term 1:   2
Enter the coefficient for term 2:  5
Enter the exponent for term 2:  1

(3.0x^2)
```



```
void print(struct node* head)
{
    if(head == NULL)
        printf("No Polynomial.");
    else {
        struct node* temp = head;
        while(temp != NULL)
        {
            printf("(%.1fx^%d)", temp->coeff, temp->expo);
            temp = temp->link;
            if(temp!=NULL)
                printf(" + ");
            else
                printf("\n");
        }
    }
}
```

Console Window

```
Enter the polynomial
Enter the number of terms:  2
Enter the coefficient for term 1:  3
Enter the exponent for term 1:   2
Enter the coefficient for term 2:  5
Enter the exponent for term 2:  1

(3.0x^2)
```



```
void print(struct node* head)
{
    if(head == NULL)
        printf("No Polynomial.");
    else {
        struct node* temp = head;
        while(temp != NULL)
        {
            printf("(%.1fx^%d)", temp->coeff, temp->expo);
            temp = temp->link;
            if(temp!=NULL)
                printf(" + ");
            else
                printf("\n");
        }
    }
}
```

Console Window

```
Enter the polynomial
Enter the number of terms:  2
Enter the coefficient for term 1:  3
Enter the exponent for term 1:   2
Enter the coefficient for term 2:  5
Enter the exponent for term 2:  1

(3.0x^2) + (5.0x^1)
```

Courtesy : Neso Academy