

**Data Structures Laboratory**  
**Assignment-2**

**Polynomial Manipulation using ListADT**

Name: Sai Shashaank R

Register No: 205001086

Class: CSE-B

**adt.h**

```
struct polyADT
{
    int coeff;
    int exp;
    struct polyADT *next;
};

struct polyADT* insertEnd(struct polyADT *p, int coeff, int exp);
//To create polynomial through insertion at the end

struct polyADT* polyAdd(struct polyADT *p1, struct polyADT *p2);
//To add two polynomials and return the pointer to the resulting
polynomial

struct polyADT* polyMul(struct polyADT *p1, struct polyADT *p2);
//To multiply two polynomials and return the pointer to the
resulting polynomial

struct polyADT* polySimplify(struct polyADT *p); //To simplify a
polynomial

int polyEvaluate(struct polyADT *p, int x); //To evaluate a
polynomial

int polyDegree(struct polyADT *p); //To find the degree of a
polynomial

void display(struct polyADT *p); //To display a listADT
implementation of polynomial

int isZero(struct polyADT *p); //To check if a polynomial is zero

int item_in(int arr[],int size,int item); //To check if an item
exists in an array; used in polySimplify()

struct polyADT* polyInp(struct polyADT *p); //To take input for a
polynomial
```

## impl.h

```
#include "adt.h"
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

struct polyADT* insertEnd(struct polyADT *p, int c, int e)
{
    struct polyADT *temp;
    temp = (struct polyADT *) malloc(sizeof(struct polyADT));
    temp->coeff = c;
    temp->exp = e;
    temp->next = NULL;

    if(p==NULL)
    {
        p=temp;
    }
    else
    {
        struct polyADT *i = p;
        while(i->next!=NULL)
            i=i->next;
        i->next = temp;
    }
    return p;
}

struct polyADT* polyAdd(struct polyADT *p1, struct polyADT *p2)
{
    if(isZero(p1))
        return p2;

    else if(isZero(p2))
```

```

        return p1;

struct polyADT *p = NULL;

while(p1!=NULL && p2!=NULL)
{
    if((p1->exp) > (p2->exp))
    {
        p = insertEnd(p,p1->coeff,p1->exp);
        p1=p1->next;
    }

    else if((p1->exp) == (p2->exp))
    {
        p = insertEnd(p,(p1->coeff + p2->coeff),p1->exp);
        p1 = p1->next;
        p2 = p2->next;
    }

    else if((p1->exp) < (p2->exp))
    {
        p = insertEnd(p,p2->coeff,p2->exp);
        p2=p2->next;
    }
}

if(p1 == NULL)
{
    while(p2!=NULL)
    {
        p = insertEnd(p,p2->coeff,p2->exp);
        p2 = p2->next;
    }
}

```

```

    if(p2==NULL)
    {
        while(p1!=NULL)
        {
            p = insertEnd(p,p1->coeff,p1->exp);
            p1 = p1->next;
        }
    }
    return p;
}

struct polyADT* polyMul(struct polyADT *p1, struct polyADT *p2)
{
    if(isZero(p1)||isZero(p2))
        return NULL;

    struct polyADT *p = NULL,*t;

    while(p1!=NULL)
    {
        t = p2;
        while(t!=NULL)
        {
            p = insertEnd(p,(p1->coeff * t->coeff),(p1->exp + t->exp));
            t = t->next;
        }

        p1=p1->next;
    }

    return p;
}

struct polyADT* polySimplify(struct polyADT *p)
{

```

```

struct polyADT *temp;
struct polyADT *p1 = NULL;

int c[50];
int e[50];

int k=0;
int result;

while(p!=NULL)
{
    result = 0;
    if(!item_in(e,k,p->exp))
    {
        result += p->coeff;
        temp = p->next;

        while(temp!=NULL)
        {
            if(temp->exp == p->exp)
                result += temp->coeff;
            temp = temp -> next;
        }

        e[k] = p->exp;
        c[k++] = result;
    }

    p = p->next;
}

for(int i=0; i<k ; i++)
{
    p1 = insertEnd(p1,c[i],e[i]);
}

```

```

        return p1;
    }

int polyEvaluate(struct polyADT *p, int x)
{
    int result=0;
    while(p!=NULL)
    {
        result += p->coeff * (int)pow(x,p->exp);
        p = p->next;
    }

    return result;
}

```

```

int polyDegree(struct polyADT *p)
{
    int largest = p->exp;
    while(p!=NULL)
    {
        if(largest < p->exp)
            largest = p->exp;
        p = p->next;
    }

    return largest;
}

```

```

int item_in(int arr[],int size,int item)
{
    int flag=0;
    for(int i=0;i<size;i++)
    {
        if(arr[i] == item)

```

```

        {
            flag = 1;
            break;
        }
    }

    return flag;
}

int isZero(struct polyADT *p)
{
    int flag=0;
    int onlyZero = (p->next == NULL)&&(p->coeff == 0);

    if(p == NULL || onlyZero)
        flag=1;

    return flag;
}

struct polyADT* polyInp(struct polyADT *p)
{
    char choice;
    int term =1;
    int c,e;
    do
    {
        printf("\nEnter coeff and exp for term %d: ",term++);
        scanf("%d %d",&c,&e);
        p = insertEnd(p,c,e);

        printf("\nDo you want to add another term?(y/n): ");
        scanf(" %c",&choice);

    }while(choice == 'y');
}

```

```
        return p;
    }

void display(struct polyADT *p)
{
    struct polyADT *ptr = p;
    while(ptr!=NULL)
    {
        printf("%dx^%d + ", ptr->coeff, ptr->exp);
        ptr=ptr->next;

    }
    printf("\b\b");
}
```



## appl.c

```
#include "impl.h"
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int choice;
    int poly_option;
    int x;

    char option;

    struct polyADT *p1 = NULL;
    struct polyADT *p2 = NULL;
    struct polyADT *p = NULL;

    printf("\nEnter polynomial 1: ");
    p1 = polyInp(p1);

    printf("\nEnter polynomial 2: ");
    p2 = polyInp(p2);

    do
    {
        printf("\nMAIN MENU");
        printf("\n1)Add two polynomials. ");
        printf("\n2)Multiply two polynomials. ");
        printf("\n3)Print the degree of the polynomials. ");
        printf("\n4)Evaluate a polynomial. ");
        printf("\n5)Exit");
        printf("\nEnter your choice: ");
        scanf("%d",&choice);

        switch(choice)
```

```

{
    case 1: p = polyAdd(p1,p2);
        printf ("\nAddition result: ");
        display(p);
        break;

    case 2: p = polyMul(p1,p2);
        p = polySimplify(p); //Using polySimplify() to
simplify the multiplication result

        /*polyMul returns NULL if the result is 0. We
need to check for that.*/
        if(p == NULL)
            printf("\nMultiplication result = 0");

        else
        {
            printf ("\nMultiplication result: ");
            display(p);
        }

        break;

    case 3: printf("\nDegree of poly1 =
%d",polyDegree(p1));
        printf("\nDegree of poly2 =
%d",polyDegree(p2));
        break;

    case 4: printf("\nEnter the polynomial to be
evaluated (1/2): ");
        scanf("%d",&poly_option);

        printf("\nEnter the value of x: ");
        scanf("%d",&x);

        if(poly_option == 1)

```

```

        printf("\nValue of poly 1 at x = %d is
%d",x,polyEvaluate(p1,x));

        else if(poly_option == 2)
            printf("\nValue of poly 2 at x = %d is
%d",x,polyEvaluate(p2,x));

        else
        {
            printf("\nERROR: Wrong choice. Try
again.");
            continue;
        }

        break;

    case 5: exit(0);

    default: printf("\nERROR. Wrong choice. Try
again.");

}

printf("\nDo you want to continue? (y/n): ");
scanf(" %c",&option);

}while(option == 'y');
}

```

## Screenshots(general cases)

1) Taking input for the two polynomials.

```
sai@sai-HP-Laptop-15q-bu0xx: ~/Documents/dsa/poly
sai@sai-HP-Laptop-15q-bu0xx:~/Documents/dsa/poly$ gcc appl.c -o appl.o -lm
sai@sai-HP-Laptop-15q-bu0xx:~/Documents/dsa/poly$ ./appl.o

Enter polynomial 1:
Enter coeff and exp for term 1: 5 2

Do you want to add another term?(y/n): y

Enter coeff and exp for term 2: 4 1

Do you want to add another term?(y/n): y

Enter coeff and exp for term 3: 3 0

Do you want to add another term?(y/n): n

Enter polynomial 2:
Enter coeff and exp for term 1: 6 1

Do you want to add another term?(y/n): y

Enter coeff and exp for term 2: 5 0

Do you want to add another term?(y/n): n

MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: █
```

2) Addition

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 1

Addition result: 5x^2 + 10x^1 + 8x^0 +
Do you want to continue? (y/n): y█
```

### 3) Multiplication

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 2

Multiplication result: 30x^3 + 49x^2 + 38x^1 + 15x^0 +
Do you want to continue? (y/n): y
```

### 4) Printing degree of the two polynomials.

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 3

Degree of poly1 = 2
Degree of poly2 = 1
Do you want to continue? (y/n): y
```

4)Evaluation of a polynomial.

a) First polynomial  $5x^2 + 4x + 3$  at  $x=2$ .

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 4

Enter the polynomial to be evaluated (1/2): 1

Enter the value of x: 2

Value of poly 1 at x = 2 is 31
Do you want to continue? (y/n): y
```

b) Second polynomial  $6x + 5$  at  $x=4$ .

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 4

Enter the polynomial to be evaluated (1/2): 2

Enter the value of x: 4

Value of poly 2 at x = 4 is 29
Do you want to continue? (y/n): y
```

c) Wrong option

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 4

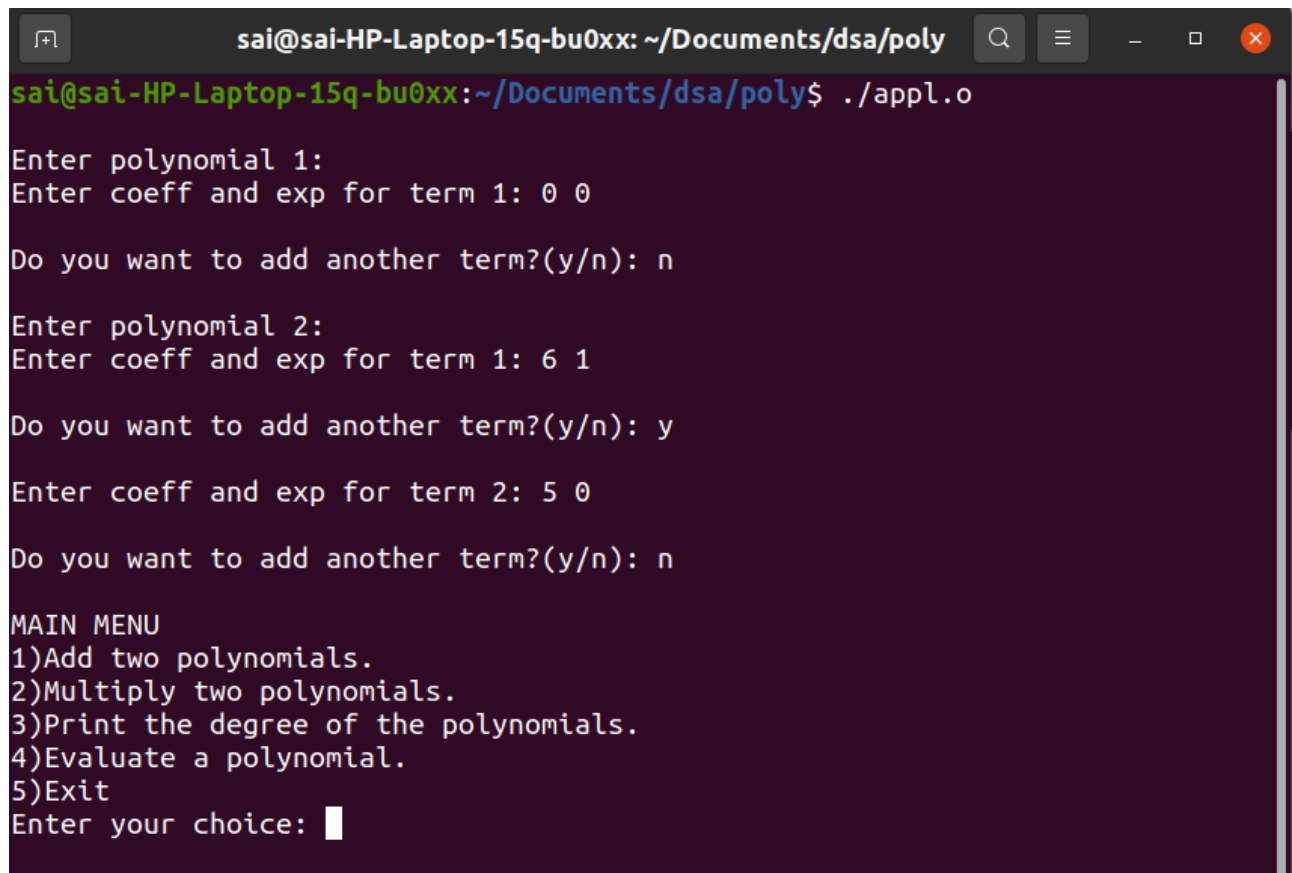
Enter the polynomial to be evaluated (1/2): 3

Enter the value of x: 2

ERROR: Wrong choice. Try again.
```

## Screenshots(when polynomial 1 = 0)

1) Taking input for the two polynomials.

A terminal window with a dark background and light-colored text. The window title is 'sai@sai-HP-Laptop-15q-bu0xx: ~/Documents/dsa/poly'. The prompt is 'sai@sai-HP-Laptop-15q-bu0xx:~/Documents/dsa/poly\$'. The user has entered './appl.o'. The program prompts for polynomial 1, then for coefficients and exponents of terms. The user enters '0 0' for the first term and 'n' for 'Do you want to add another term?(y/n)'. Then it prompts for polynomial 2, and the user enters '6 1' for the first term and 'y' for 'Do you want to add another term?(y/n)'. Then it prompts for coefficients and exponents of terms for polynomial 2, and the user enters '5 0' and 'n' for 'Do you want to add another term?(y/n)'. Finally, it displays a 'MAIN MENU' with five options: 1)Add two polynomials., 2)Multiply two polynomials., 3)Print the degree of the polynomials., 4)Evaluate a polynomial., 5)Exit. The prompt 'Enter your choice:' is followed by a cursor.

```
sai@sai-HP-Laptop-15q-bu0xx: ~/Documents/dsa/poly$ ./appl.o
Enter polynomial 1:
Enter coeff and exp for term 1: 0 0
Do you want to add another term?(y/n): n
Enter polynomial 2:
Enter coeff and exp for term 1: 6 1
Do you want to add another term?(y/n): y
Enter coeff and exp for term 2: 5 0
Do you want to add another term?(y/n): n
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: █
```

## 2) Addition

The second polynomial is printed as the result.

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 1

Addition result: 6x^1 + 5x^0 +
Do you want to continue? (y/n): y
```

## 3) Multiplication

The result is printed as zero.

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 2

Multiplication result = 0
Do you want to continue? (y/n): y
```

## 4) Degree of the two polynomials

```
MAIN MENU
1)Add two polynomials.
2)Multiply two polynomials.
3)Print the degree of the polynomials.
4)Evaluate a polynomial.
5)Exit
Enter your choice: 3

Degree of poly1 = 0
Degree of poly2 = 1
Do you want to continue? (y/n): n
```