Date ___ / ___ / _____

Addition of two long Integers

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct node
{
    int info;
    struct NODE * link;
};
typedef struct NODE * node;
node getnode()
{
  node x;
  x = (node) malloc(sizeof(struct NODE));
if(x == NULL)
  {
    printf("out of memory\n");
    exit(0);
  }
return x;
}
node ins_front(node first, int item)
 {
    node temp;
    temp = getnode();
    temp -> info = item;
    temp -> link = first;
    return temp;
}
node extract(char *s, node head)
```

Date ___/___/_____

```
{
  int i, n;
  for(i=0; i<strlen(s); i++)
  {
    n = s[i] - '0';
    head = ins_front(head, n);
  }
  return head;
}
node addlong(node head1, node head2, node head3)
{
  int temp, sum, carry = 0;
  node cur1, cur2;
  cur1 = head1;
  cur2 = head2;
  while(cur1 != NULL && cur2 != NULL)
  {
    temp = cur1 -> info + cur2 -> info + carry;
    if(temp > 9)
    {
      sum = temp % 10;
      carry = temp / 10;
    }
    else
    {
      sum = temp;
      carry = 0;
    }
  head3 = ins_front(head3, sum);
  cur1 = cur1 -> link;
  cur2 = cur2 -> link;
```

Page No.

```
}
while (cur1 != NULL)
{
    temp = cur1 -> info + carry;
    if (temp > 9)
    {
        sum = temp % 10;
        carry = temp / 10;
    }
    else
    {
        sum = temp;
        carry = 0;
    }
    head3 = ins_front (head3, sum);
    cur1 = cur1 -> link;
}
while (cur2 != NULL)
{
    temp = cur2 -> info + carry;
    if (temp > 9)
    {
        sum = temp % 10;
        carry = temp / 10;
    }
    else
    {
        sum = temp;
        carry = 0;
    }
    head3 = ins_front (head3, sum);
    cur2 = cur2 -> link;
}
```

```
if(cur1 == NULL && cur2 == NULL)
{
    if (carry == 1)
    head3 = ins_front(head3, carry);
}
return head3;
}
void display (node first)
{
    node cur;
    if (first == NULL)
    {
        printf("Empty \n");
        return;
    }
    cur = first;
    while (cur != NULL)
    {
        temp == cur2 -> info + carry;
        if printf("%d \t", cur -> info);
        cur = cur -> link;
    }
}
void main()
{
    int ch;
    node head1 = NULL;
    node head2 = NULL;
    node head3 = NULL;
    Char S1[30], S2[30];
    printf("\n Enter first integer \n");
```

Date ___/___/_____

```
scanf("%s", s1);
head1 = extract(s1, head1);
display(head1);
printf("\n Enter second integer \n");
scanf("%s", s2);
head2 = extract(s2, head2);
display(head2);
head3 = addlong(head1, head2, head3);
printf("\n The result is \n");
display(head3);
}
```

Date __/__/____

Addition of Two Polynomials

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
struct node
{
    float cf;
    float px;
    float py;
    int flag;
    struct node * link;
};
typedef struct node * NODE;
NODE getnode()
{
  NODE x;
  x = (NODE) malloc (size of (struct node));
  if (x == NULL)
  {
    printf("out of memory \n");
    exit(0);
  }
  return x;
}
NODE insert_rear (float cf, float x, float y,
                  NODE head)
{
  NODE temp, cur;
  temp = getnode();
  temp -> cf = cf;
  temp -> px = x;
```

```
    temp -> py = y;
    temp -> cf = 0;
    cur = head -> link;
while (cur -> link != head)
cur = cur -> link;
cur -> link = temp;
 temp -> link = head;
return head;
}
void display (NODE head)
{
    NODE temp;
  if (head -> link == head)
  {
  printf ("Polynomial does not exist \n");
  return;
  }
  temp = head -> link;
  while (temp != head)
  {
    printf ("+%5.2fx^%3.1fy^%3.1f",
                    temp -> cf, temp -> px, temp -> py);
  temp = temp -> link;
  }
  printf ("\n");
}
NODE add_poly (NODE h1, NODE h2, NODE h3)
{
  NODE p1, p2;
  int x1, x2, y1, y2, cf1, cf2, cf;
  p1 = h1 -> link;
  while (p1 != h1)
```

Date ___/___/_____

```
{
    x1 = p1 -> px;
    y1 = p1 -> py;
    cf1 = p1 -> cf;
    p2 = h2 -> link;
    while (p2 != h2)
    {
        x2 = p2 -> px;
        y2 = p2 -> py;
        cf2 = p2 -> cf;
        if (x1 == x2 && y1 == y2)
        break;
        p2 = p2 -> link;
    }
    if (p2 != h2)
    {
        cf = cf1 + cf2;
        p2 -> flag = 1;
        if (cf != 0)
        h3 = insert_rear (cf, x1, y1, h3);
    }
    else
    h3 = insert_rear (cf1, x1, y1, h3);
    p1 = p1 -> link;
    }
    p2 = h2 -> link;
    while (p2 != h2)
    {
        if (p2 -> flag == 0)
        h3 = insert_rear (p2 -> cf, p2 -> px,
                          p2 -> py, h3);
    }
```

Page No.

```
    p2 = p2 -> link;
  }
  return h3;
}
NODE read_poly(NODE head)
{
  int i;
  float px;
  float py;
  float cf;
  printf("Enter the coefficients as -999 to
                    end the polynomial \n");
  for(i=1;;i++)
  {
    printf("enter the %d term \n", i);
    printf("coeff = ");
    scanf("%f", &cf);
    if(cf == -999)
    break;
    printf("power of x = ");
    scanf("%f", &px);
    printf("power of y = ");
    scanf("%f", &py);
    head = insert_rear(cf, px, py, head);
  }
  return head;
}
void main()
{
  NODE h1, h2, h3;
  h1 = getnode();
  h2 = getnode();
```

Date __/__/____

```
h3 = getnode();
h1 -> link = h1;
h2 -> link = h2;
h3 -> link = h3;
printf("Enter the first polynomial : \n");
h1 = read_poly(h1);
printf("Enter the second polynomial : \n");
h2 = read_poly(h2);
h3 = add_poly(h1, h2, h3);
printf("The first polynomial is : \n");
display(h1);
printf("The second polynomial is : \n");
display(h2);
printf("The sum of two polynomial is : \n");
display(h3);
}
```

Date __/__/____

Evaluation of a polynomial

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
struct node
{
    float cf;
    float px;
    float py;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x = (NODE) malloc (sizeof (struct node));
    if (x==NULL)
    {
        printf("out of memory\n");
        exit(0);
    }
    return x;
}
NODE insert_rear(float cf, float x, float y, NODE head)
{
    NODE temp, cur;
    temp = getnode();
    temp->cf = cf;
```

Date ___/___/_____

```
temp->px = x;
temp->py = y;
cur = head->link;
while (cur->link != head)
cur = cur->link;
cur->link = temp;
temp->link = head;
return head;
}
NODE read_poly (NODE head)
{
  int i;
  float cf, px, py;
  printf ("Enter the coefficients as -999 to end
       the polynomial \n");
  for (i = 1; ; i++)
  {
    printf ("Enter the %d term \n", i);
    printf ("Coeff = ");
    scanf ("%f", &cf);
    if (cf == -999)
    break;
    printf ("pow x = ");
    scanf ("%f", &px);
    printf ("pow y = ");
    scanf ("%f", &py);
    head = insert_rear (cf, px, py, head);
  }
float evaluate (NODE head)
{
    float sum=0, x, y;
    NODE poly;
```

```
printf("Enter the value of x and y \n");
scanf("%f %f", &x, &y);
poly = head -> link;
while (poly != head)
{
sum = sum + poly -> cf * pow(x, poly -> px)
                  * pow(y, poly -> py);
poly = poly -> link;
}
return sum;
}
void display (NODE head)
{
  NODE temp;
  if(head -> link == head)
  {
  printf("polynomial does not exist\n");
    return;
  }
temp = head -> link;
while ( temp != head)
{
printf("+%.5.2fx^%.3.1fy^%.3.1f", temp -> cf,
                                  temp -> px, temp -> py);
  temp = temp -> link;
}
printf("\n");
}
void main()
{
  NODE head;
```

Date __/__/____

```
float res;
head = getnode();
head->link = head;
printf("Enter the polynomial\n");
head = read_poly(head);
printf("The given polynomial is \n");
display(head);
printf("The result is %f \n", res);
}
```