Lab Assignment 4-Multi Precision Arithmetic

CODE :

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int number;

    struct node \*next;

};

struct node \*create(struct node \*head);

struct node \*insert(struct node \*head, int n);

struct node \*Add(struct node \*head1, struct node \*head2, struct node \*result);

struct node \*resultt(struct node \*result, int sum);

void display(struct node \*result);

int main()

{

    struct node \*head1 = NULL, \*head2 = NULL, \*result = NULL;

    printf("Number 1 : \n");

    head1 = create(head1);

    printf("Number 2 : \n");

    head2 = create(head2);

    result = Add(head1, head2, result);

    printf("Result : \n");

    display(result);

    return 0;

}

struct node \*create(struct node \*head)

{

    int n;

    printf("Enter the number of the  digits in the  number : ");

    scanf("%d", &n);

    char A[n], R[n];

    printf("Enter the number: ");

    scanf("%s", A);

    for (int i = n - 1, j = 0; i >= 0 && j < n; i--, j++)

    {

        R[j] = A[i];

    }

    int i = 0;

    while (i < n)

    {

        head = insert(head, R[i] - '0');

        i++;

    }

    return head;

}

struct node \*insert(struct node \*head, int n)

{

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    struct node \*temp = head;

    ptr->number = n;

    ptr->next = NULL;

    if (head == NULL)

    {

        head = ptr;

    }

    else

    {

        while (temp->next != NULL)

        {

            temp = temp->next;

        }

        temp->next = ptr;

    }

    return head;

}

struct node \*Add(struct node \*head1, struct node \*head2, struct node \*result)

{

    int carry = 0;

    int sum = 0;

    struct node \*temp1 = head1, \*temp2 = head2;

    while (temp1 != NULL && temp2 != NULL)

    {

        sum = temp1->number + temp2->number + carry;

        carry = 0;

        carry = sum / 10;

        sum = sum % 10;

        result = resultt(result, sum);

        temp1 = temp1->next;

        temp2 = temp2->next;

    }

    if (temp1 == NULL && temp2 == NULL)

    {

        if (carry != 0)

            result = resultt(result, carry);

    }

    sum = 0;

    while (temp1 != NULL)

    {

        sum = temp1->number + carry;

        carry = 0;

        carry = sum / 10;

        sum = sum % 10;

        result = resultt(result, sum);

        temp1 = temp1->next;

    }

    while (temp2 != NULL)

    {

        sum = temp2->number + carry;

        carry = 0;

        carry = sum / 10;

        sum = sum % 10;

        result = resultt(result, sum);

        temp2 = temp2->next;

    }

    return result;

}

struct node \*resultt(struct node \*result, int sum)

{

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    ptr->number = sum;

    if (result == NULL)

    {

        result = ptr;

        ptr->next = NULL;

    }

    else

    {

        ptr->next = result;

        result = ptr;

    }

    return result;

}

void display(struct node \*result)

{

    struct node \*temp = result;

    while (temp != NULL)

    {

        printf("%d", temp->number);

        temp = temp->next;

    }

}

OUTPUT :

Number 1 :

Enter the number of the digits in the number : 4

Enter the number: 2564

Number 2 :

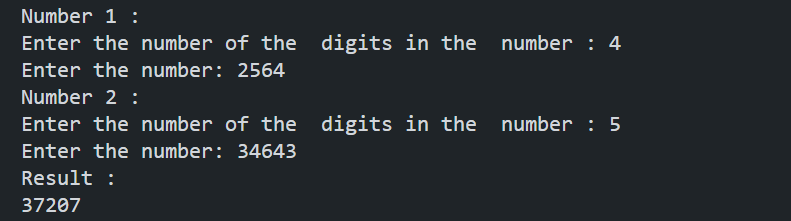
Enter the number of the digits in the number : 5

Enter the number: 34643

Result :

37207

SCREENSHOT :



TIME COMPLEXITY :

Create function’s time complexity O(n^2)

Dominates so ,

Time complexity of the code :

O(n^2)