

Assignment 3

Name : Om D. Tambat

Enrollment No. : BT22CSH042

Q1.

Code:

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

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

```
struct Node {
```

```
    int row;
```

```
    int col;
```

```
    int value;
```

```
    struct Node* next;
```

```
};
```

```
struct Node* createNode(int row, int col, int value) {
```

```
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
```

```
    if (newNode == NULL) {
```

```
        printf("Memory allocation failed.\n");
```

```
        exit(1);
```

```
    }
```

```
    newNode->row = row;
```

```
    newNode->col = col;
```

```
    newNode->value = value;
```

```
    newNode->next = NULL;
```

```
    return newNode;
```

```
}
```

```
void displaySparseMatrix(struct Node* head) {  
    struct Node* current = head;  
  
    printf("Row\tColumn\tValue\n");  
    while (current != NULL) {  
        printf("%d\t%d\t%d\n", current->row, current->col, current->value);  
        current = current->next;  
    }  
}
```

```
int main() {  
    int rows, cols;  
    printf("Enter the number of rows and columns of the matrix: ");  
    scanf("%d %d", &rows, &cols);  
  
    struct Node* head = NULL;  
  
    for (int i = 0; i < rows; i++) {  
        for (int j = 0; j < cols; j++) {  
            int value;  
            printf("Enter the value at row %d, column %d: ", i, j);  
            scanf("%d", &value);  
  
            if (value != 0) {
```

```
    struct Node* newNode = createNode(i, j, value);
    if (head == NULL) {
        head = newNode;
    } else {
        newNode->next = head;
        head = newNode;
    }
}
}
```

```
printf("Linked List Representation of Sparse Matrix:\n");
displaySparseMatrix(head);
```

```
while (head != NULL) {
    struct Node* temp = head;
    head = head->next;
    free(temp);
}
```

```
    return 0;
}
```

Output :

Output

Clear

```
/tmp/lBckBXXUgn.o
Enter the number of rows and columns of the matrix: 2
2
Enter the value at row 0, column 0: 1
Enter the value at row 0, column 1: 0
Enter the value at row 1, column 0: 0
0Enter the value at row 1, column 1: 0
Linked List Representation of Sparse Matrix:
Row Column Value
0 0 1
|
```

Q2.

Code:

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

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

```
struct Node {
    int data;
    struct Node* next;
};
```

```
typedef struct Node Node;
```

```
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    if (newNode == NULL) {
        printf("Memory allocation failed.\n");
        exit(1);
    }
}
```

```
newNode->data = data;
newNode->next = NULL;
return newNode;
}
```

```
void insertEnd(Node** head, int data) {
    Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}
```

```
Node* reverseList(Node* head) {
    Node* prev = NULL;
    Node* current = head;
    Node* next = NULL;
    while (current != NULL) {
        next = current->next;
        current->next = prev;
        prev = current;
```

```
        current = next;
    }
    return prev;
}
```

```
Node* addLongIntegers(Node* num1, Node* num2) {
    Node* result = NULL;
    int carry = 0;

    while (num1 != NULL || num2 != NULL || carry != 0) {
        int sum = carry;
        if (num1 != NULL) {
            sum += num1->data;
            num1 = num1->next;
        }
        if (num2 != NULL) {
            sum += num2->data;
            num2 = num2->next;
        }

        carry = sum / 10;
        sum = sum % 10;

        insertEnd(&result, sum);
    }
}
```

```
        return reverseList(result);
    }

void displayList(Node* head) {
    Node* current = head;
    while (current != NULL) {
        printf("%d", current->data);
        current = current->next;
    }
    printf("\n");
}
```

```
int main() {
    Node* num1 = NULL;
    Node* num2 = NULL;

    char num1Str[100];
    printf("Enter the first long integer: ");
    scanf("%s", num1Str);
    for (int i = 0; num1Str[i] != '\0'; i++) {
        insertEnd(&num1, num1Str[i] - '0');
    }

    char num2Str[100];
    printf("Enter the second long integer: ");
    scanf("%s", num2Str);
```

```
for (int i = 0; num2Str[i] != '\0'; i++) {  
    insertEnd(&num2, num2Str[i] - '0');  
}
```

```
Node* result = addLongIntegers(num1, num2);  
printf("Sum of the two long integers: ");  
displayList(result);
```

```
while (num1 != NULL) {  
    Node* temp = num1;  
    num1 = num1->next;  
    free(temp);  
}
```

```
while (num2 != NULL) {  
    Node* temp = num2;  
    num2 = num2->next;  
    free(temp);  
}
```

```
while (result != NULL) {  
    Node* temp = result;  
    result = result->next;  
    free(temp);  
}
```



```
    return 0;  
}
```

Output :

Output

Clear

```
/tmp/JHwNrqbIps.o  
Enter the first long integer: 836251  
Enter the second long integer: 729480  
Sum of the two long integers: 237565
```