WEEK 8:

1.SORTING A LINKED LIST ,REVERSING AND CONCATENATE:

#include <stdio.h>

#include <stdlib.h>

struct Node {

    int data;

    struct Node\* next;

};

struct Node\* createNode(int data) {

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

    newNode->data = data;

    newNode->next = NULL;

    return newNode;

}

void append(struct Node\*\* head, int data) {

    struct Node\* newNode = createNode(data);

    if (\*head == NULL) {

        \*head = newNode;

        return;

    }

    struct Node\* temp = \*head;

    while (temp->next != NULL) {

        temp = temp->next;

    }

    temp->next = newNode;

}

void display(struct Node\* head) {

    if (head == NULL) {

        printf("Linked list is empty.\n");

        return;

    }

    struct Node\* temp = head;

    while (temp != NULL) {

        printf("%d -> ", temp->data);

        temp = temp->next;

    }

    printf("NULL\n");

}

void reverse(struct Node\*\* head) {

    struct Node\* prev = NULL;

    struct Node\* current = \*head;

    struct Node\* next = NULL;

    while (current != NULL) {

        next = current->next;

        current->next = prev;

        prev = current;

        current = next;

    }

    \*head = prev;

}

void bubbleSort(struct Node\* head) {

    if (head == NULL) return;

    struct Node\* current;

    struct Node\* lastSorted = NULL;

    int swapped;

    do {

        swapped = 0;

        current = head;

        while (current->next != lastSorted) {

            if (current->data > current->next->data) {

                int temp = current->data;

                current->data = current->next->data;

                current->next->data = temp;

                swapped = 1;

            }

            current = current->next;

        }

        lastSorted = current;

    } while (swapped);

}

void concatenate(struct Node\*\* head1, struct Node\*\* head2) {

    if (\*head1 == NULL) {

        \*head1 = \*head2;

        return;

    }

    struct Node\* temp = \*head1;

    while (temp->next != NULL) {

        temp = temp->next;

    }

    temp->next = \*head2;

}

int main() {

    struct Node\* list1 = NULL;

    struct Node\* list2 = NULL;

    printf("Appending elements to the first linked list:\n");

    append(&list1, 33);

    append(&list1, 10);

    append(&list1, 56);

    display(list1);

    printf("\nSorting the first linked list:\n");

    bubbleSort(list1);

    display(list1);

    printf("\nReversing the first linked list:\n");

    reverse(&list1);

    display(list1);

    printf("\nAppending elements to the second linked list:\n");

    append(&list2, 42);

    append(&list2, 5);

    append(&list2, 69);

    display(list2);

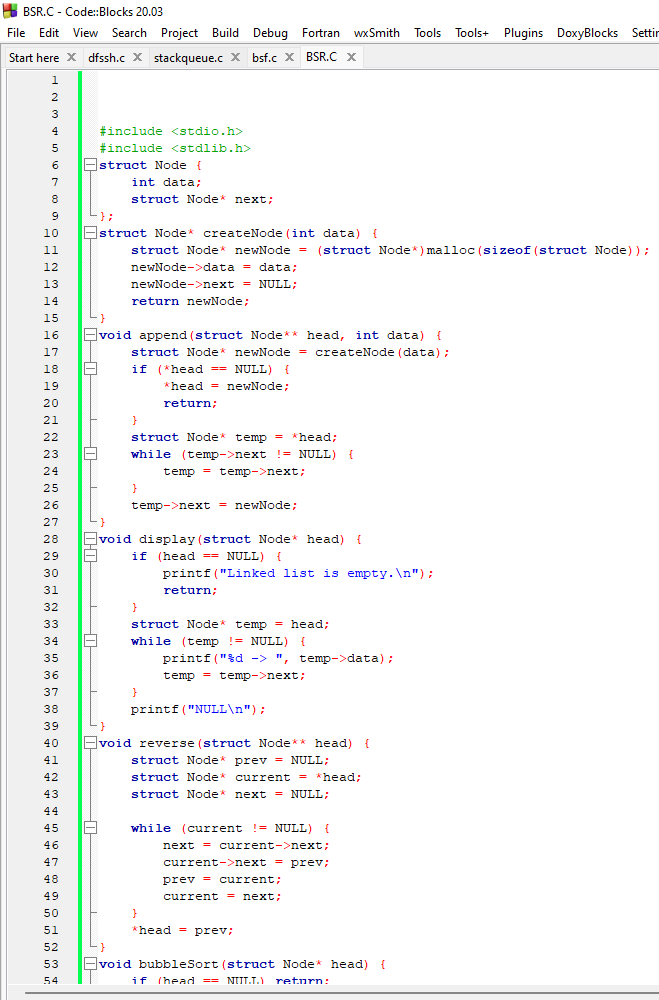
    printf("\nConcatenating the two linked lists:\n");

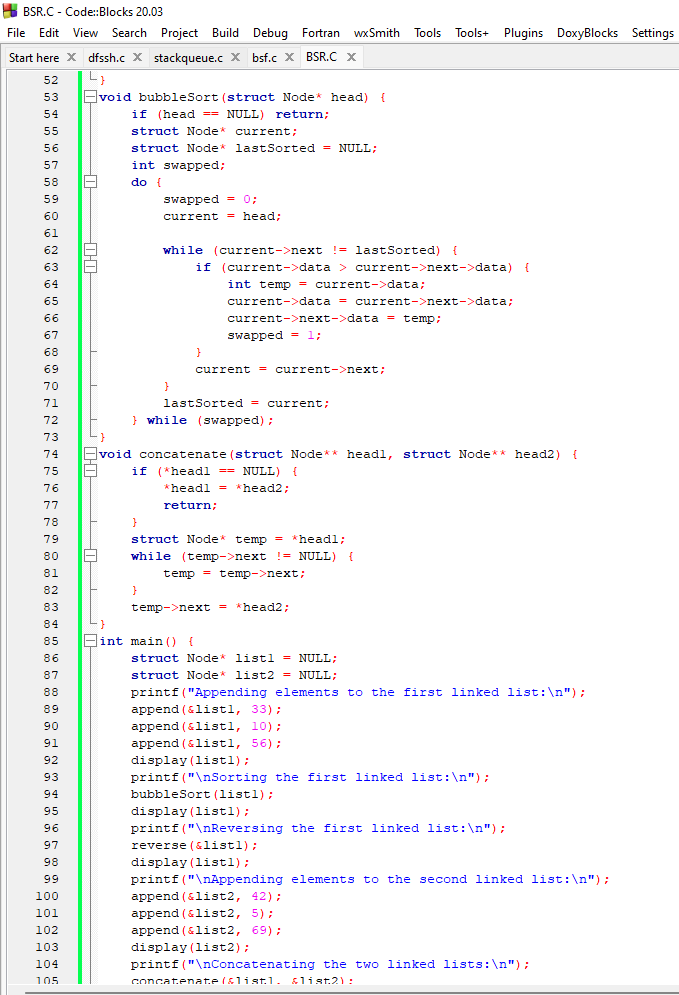
    concatenate(&list1, &list2);

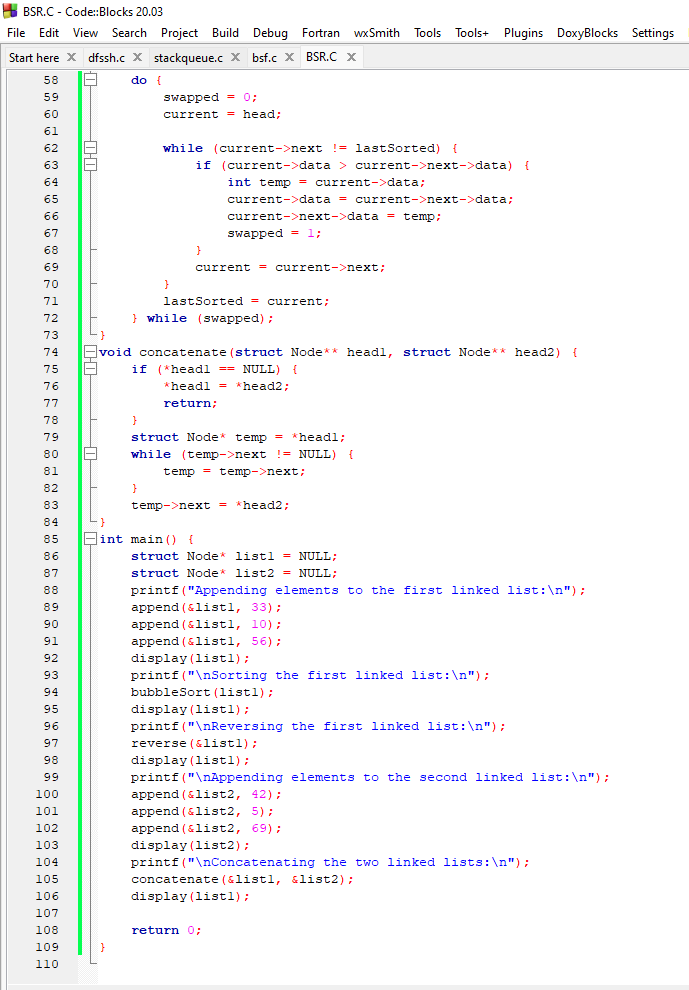
    display(list1);

    return 0;

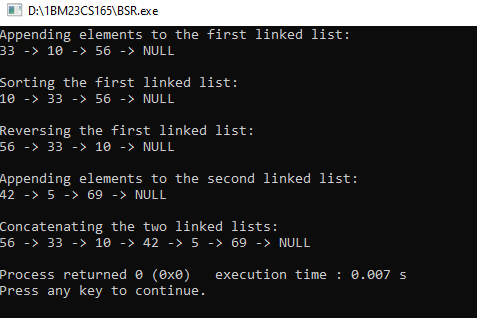
}

SSS

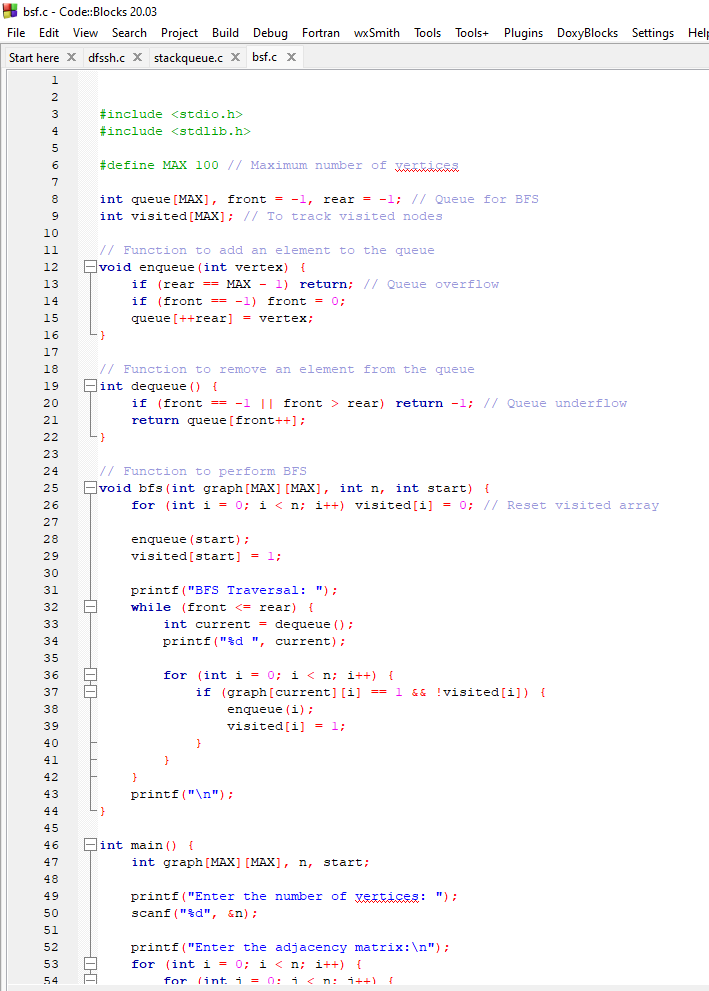
AA

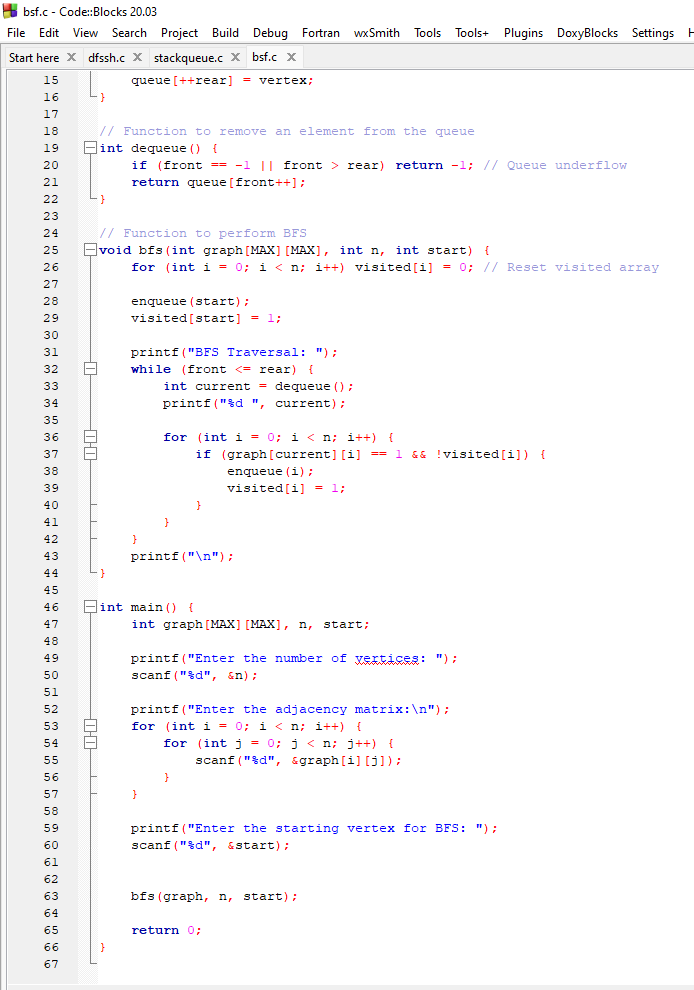
AA

OUTPUT:

A

2.STACK AND QUEUE USING LINKED LIST:

a

a

OUTPUT:

