|  |
| --- |
|  |
| EXPERIMENT 5 |
|  |
|  |
| **Vansh Sukhija**  **12112021** |
|  |

|  |
| --- |
|  |

Ans 1(a)

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* insert(*struct* Node \*\**ptr*, *int* *pos*){

*struct* Node \*new = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*p = \**ptr*;

    printf("Enter data: ");

    scanf("%s", &new->data);

    if(*pos*==0){

        new->next = \**ptr*;

        \**ptr* = new;

    }

    else{

        for(*int* i=0; i<*pos*-1; i++){

            p = p->next;

        }

        new->next = p->next;

        p->next = new;

    }

}

*void* main(){

*struct* Node \*first = create();

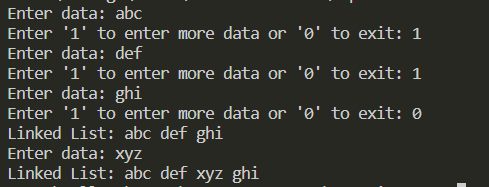
    display(first);

    insert(&first, 2);

    display(first);

}

Output-



Ans 1(b)-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*int* pos(*struct* Node \**ptr*, *char* *data*[]){

*int* count=0;

    while(*ptr*!=0){

        if(!strcmp(*ptr*->data, *data*))

            return count;

        count++;

*ptr* = *ptr*->next;

    }

    return -1;

}

*void* main(){

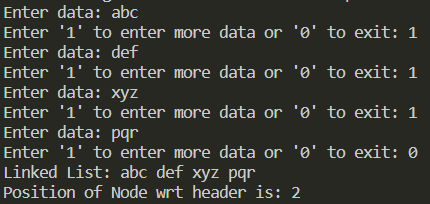
*struct* Node \*first = create();

    display(first);

    printf("Position of Node wrt header is: %d", pos(first, "xyz"));

}

Output-



Ans 1(c)-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* delete(*struct* Node \*\**ptr*, *char* *key*[]){

*struct* Node \*p = \**ptr*, \*q;

    if(!strcmp((\**ptr*)->data, *key*)){

        \**ptr* = p->next;

        p->next = NULL;

        free(p);

    }

    else{

        while(p!=0 && strcmp(p->next->data, *key*)){

            p = p->next;

        }

        q = p->next;

        p->next = q->next;

        q->next = NULL;

        free(q);

    }

}

*void* main(){

*struct* Node \*first = create();

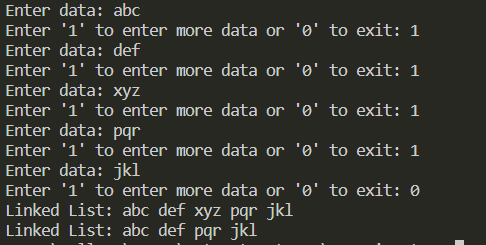
    display(first);

    delete(&first, "xyz");

    display(first);

}

Output-



Ans 1(d)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* deleteall(*struct* Node \*\**ptr*){

*struct* Node \*p = \**ptr*;

    while(p->next!=0){

        \**ptr* = p->next;

        p->next = NULL;

        free(p);

        p = \**ptr*;

    }

    \**ptr* = NULL;

    free(*ptr*);

}

*void* main(){

*struct* Node \*first = create();

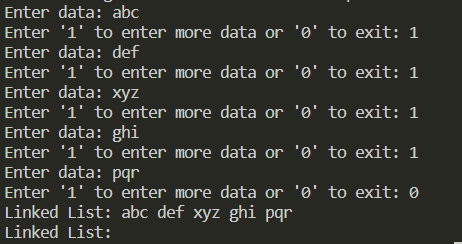
    display(first);

    deleteall(&first);

    display(first);

}

Output-



Ans 1(e)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

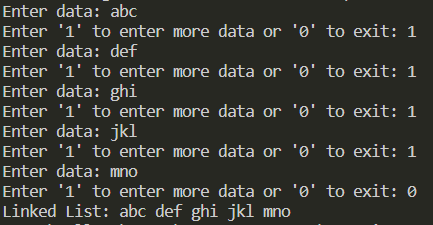
*void* main(){

*struct* Node \*first = create();

    display(first);

}

Output-



Ans 1(f)-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

*struct* Node{

*char* data[10];

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%s", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        fflush(stdin);

        printf("Enter data: ");

        scanf("%s", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%s ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*struct* Node \*copy(*struct* Node \**ptr*){

*struct* Node \*second = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    strcpy(second->data, *ptr*->data);

    second->next = NULL;

    last = second;

*ptr* = *ptr*->next;

    while(*ptr*!=0){

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

        strcpy(new->data, *ptr*->data);

        new->next = NULL;

        last->next = new;

        last = new;

*ptr* = *ptr*->next;

    }

    return second;

}

*void* main(){

*struct* Node \*first = create();

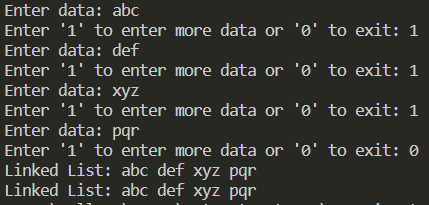
    display(first);

*struct* Node \*second = copy(first);

    display(second);

}

Output-



Ans 2(a)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*int* data;

*struct* Node \*next;

}\*first;

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%d", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        printf("Enter data: ");

        scanf("%d", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%d ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* reverse(*struct* Node \**p*, *struct* Node \**q*){

    if(*q*!=0){

        reverse(*q*, *q*->next);

*q*->next = *p*;

    }

    else

        first = *p*;

}

*void* main(){

    first = create();

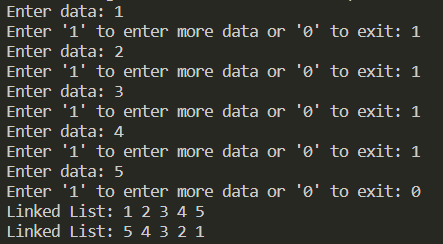
    display(first);

    reverse(0, first);

    display(first);

}

Output-



Ans 2(b)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*int* data;

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    first->next = NULL;

    last = first;

    printf("Enter data: ");

    scanf("%d", &first->data);

*int* input;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input){

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

        printf("Enter data: ");

        scanf("%d", &t->data);

        t->next = NULL;

        last->next = t;

        last = t;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%d ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* swap(*struct* Node \*\**ptr*){

*struct* Node \*p = \**ptr*, \*q = (\**ptr*)->next, \*temp = NULL;

    if(q!=0){

        p->next = q->next;

        q->next = p;

        \**ptr* = q;

    }

    temp = p;

    p = p->next;

    q = p->next;

    while(p!=0 && q!=0){

        p->next = q->next;

        q->next = p;

        temp->next = q;

        temp = p;

        p = p->next;

        q = p->next;

    }

}

*void* main(){

*struct* Node \*first = create();

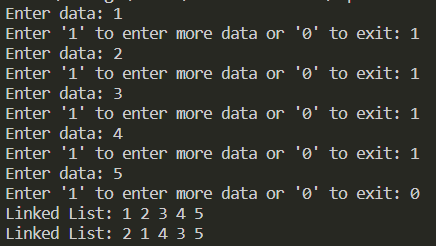
    display(first);

    swap(&first);

    display(first);

}

Output-



Ans 3(a)-

#include<stdio.h>

#include<stdlib.h>

*struct* node

{

*int* val;

*struct* node \* next;

};

*struct* stack

{

*struct* node \* top;

};

*void* create(*struct* stack \**s*){

    (*s*)->top = NULL;

}

*void* push(*struct* stack \**s*, *int* *element*){

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

    temp->val = *element*;

    temp->next = NULL;

    if(((*s*)->top)==NULL){

        (*s*)->top = temp;

    }

    else{

        temp->next = (*s*)->top;

        (*s*)->top = temp;

    }

}

*int* pop(*struct* stack \**s*){

*struct* node \* temp = (*s*)->top;

    (*s*)->top = temp->next;

*int* a = temp->val;

    free(temp);

    return a;

}

*int* isEmpty(*struct* stack \**s*){

    if((*s*)->top==NULL){

        return 1;

    }

    return 0;

}

*int* main(){

*struct* stack s;

    create(&s);

    push(&s,6);

    push(&s,7);

    push(&s,8);

    push(&s,9);

    pop(&s);

    push(&s,10);

    push(&s,11);

    printf("%d\n", pop(&s));

    printf("%d\n", pop(&s));

    printf("%d\n", pop(&s));

    push(&s,69);

    printf("%d", pop(&s));

    return 0;

}

Output-



Ans 3(b)-

#include<stdio.h>

#include<stdlib.h>

*struct* node

{

*int* val;

*struct* node \* next;

};

*struct* queue

{

*struct* node \* first, \*last;

};

*void* create(*struct* queue \* *q*){

*q*->first = *q*->last = NULL ;

}

*void* Insert(*struct* queue \* *q*, *int* *element*){

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

    temp->val = *element*;

    temp->next = NULL;

    if(*q*->first==NULL){

*q*->last = *q*->first = temp;

    }

    else{

*q*->last->next = temp;

*q*->last = temp;

    }

}

*int* Delete(*struct* queue \* *q*){

    if(*q*->first!=NULL){

*struct* node \* temp = *q*->first;

*q*->first = temp->next;

*int* a = temp->val;

        free(temp);

        return a;

    }

}

*int* isEmpty(*struct* queue \* *q*){

    if(*q*->first==NULL){

        return 1;

    }

    return 0;

}

*int* isFull(*struct* queue \* *q*){

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

    if(p!=NULL){

        free(p);

        return 0;

    }

    return 1;

}

*int* main(){

*struct* queue q;

    create(&q);

    Insert(&q,1);

    Insert(&q,2);

    Insert(&q,3);

    printf("%d\n", Delete(&q));

    Insert(&q,4);

    Insert(&q,5);

    Insert(&q,6);

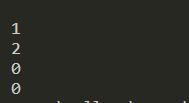
    printf("%d\n", Delete(&q));

    printf("%d\n", isEmpty(&q));

    printf("%d\n", isFull(&q));

}

Output-



Ans 4(a)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*int* coeff, expo;

*struct* Node \*next;

};

*struct* Node \*create(){

*int* n;

    printf("Enter the maximum power of 'x' in the polynomial: ");

    scanf("%d", &n);

*struct* Node \*first = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    printf("Enter the Coefficient of x^%d: ", n);

    scanf("%d", &first->coeff);

    first->expo = n;

    first->next = NULL;

    last = first;

    for(*int* i=0; i<n; i++){

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

        new->expo = n-1-i;

        printf("Enter the Coefficient of x^%d: ", new->expo);

        scanf("%d", &new->coeff);

        new->next = NULL;

        last->next = new;

        last = new;

    }

    return first;

}

*void* display(*struct* Node \**ptr*){

    printf("Polynomial: ");

    while(*ptr*!=0){

        if(*ptr*->coeff){

            printf("%dx^%d ", *ptr*->coeff, *ptr*->expo);

            if(*ptr*->next!=0 && *ptr*->next->coeff>0)

                printf("+ ");

        }

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*struct* Node \*insert(*struct* Node \**ptr*, *int* *co*, *int* *ex*){

*struct* Node \*new = malloc(sizeof(*struct* Node)), \*temp;

    new->coeff = *co*;

    new->expo = *ex*;

    new->next = NULL;

    if (*ptr* == NULL || *ex* > *ptr*->expo)    {

        new->next = *ptr*;

*ptr* = new;

    }

    else{

        temp = *ptr*;

        while (temp->next != NULL && temp->next->expo > *ex*)

            temp = temp->next;

        if(temp->next->expo == *ex*)

            temp->next->coeff += *co*;

        else{

            new->next = temp->next;

            temp->next = new;

        }

    }

    return *ptr*;

}

*struct* Node \*polyAdd(*struct* Node \**first*, *struct* Node \**second*){

*struct* Node \*third = (*struct* Node \*)malloc(sizeof(*struct* Node)), \*last;

    if(*first*->expo == *second*->expo){

        third->coeff = *first*->coeff + *second*->coeff;

        third->expo = *first*->expo;

*first* = *first*->next;

*second* = *second*->next;

    }

    else if(*first*->expo > *second*->expo){

        third->coeff = *first*->coeff;

        third->expo = *first*->expo;

*first* = *first*->next;

    }

    else{

        third->coeff = *second*->coeff;

        third->expo = *second*->expo;

*second* = *second*->next;

    }

    third->next = NULL;

    last = third;

    while(*first*!=0 && *second*!=0){

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

        new->next = NULL;

        if(*first*->expo == *second*->expo){

            new->coeff = *first*->coeff + *second*->coeff;

            new->expo = *first*->expo;

*first* = *first*->next;

*second* = *second*->next;

        }

        else if(*first*->expo > *second*->expo){

            new->coeff = *first*->coeff;

            new->expo = *first*->expo;

*first* = *first*->next;

        }

        else{

            new->coeff = *second*->coeff;

            new->expo = *second*->expo;

*second* = *second*->next;

        }

        last->next = new;

        last = new;

    }

    return third;

}

*struct* Node \*polyMul(*struct* Node \**first*, *struct* Node \**second*){

*struct* Node \*third, \*tempsecond = *second*;

    while(*first*!=0){

*second* = tempsecond;

        while(*second*!=0){

            third = insert(third, *first*->coeff\**second*->coeff, *first*->expo+*second*->expo);

*second* = *second*->next;

        }

*first* = *first*->next;

    }

    return third;

}

*void* main(){

*struct* Node \*first = create(), \*second, \*third, \*fourth;

    display(first);

    second = create();

    display(second);

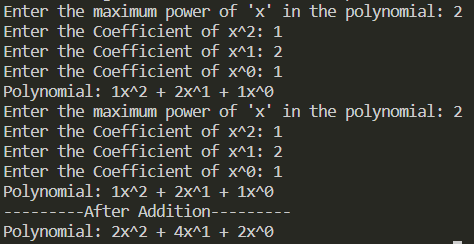
    printf("---------After Addition---------\n");

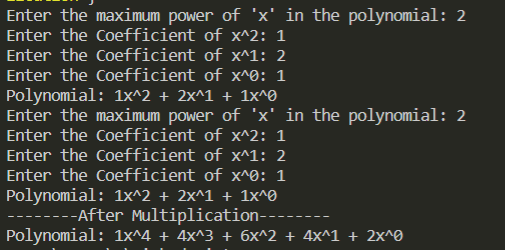
    third = polyAdd(first, second);

    display(third);

}

Output-





Ans 5(a)-

#include<stdio.h>

#include<stdlib.h>

*struct* Node{

*int* data;

*struct* Node \*next;

};

*struct* Node \*create(){

*struct* Node \*first = (*struct* Node\*)malloc(sizeof(*struct* Node)), \*last;

    printf("Enter data: ");

    scanf("%d", &first->data);

    first->next = NULL;

    last = first;

*int* input=0;

    printf("Enter '1' to enter more data or '0' to exit: ");

    scanf("%d", &input);

    while(input==1){

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

        printf("Enter data: ");

        scanf("%d", &new->data);

        new->next = NULL;

        last->next = new;

        last = new;

        printf("Enter '1' to enter more data or '0' to exit: ");

        scanf("%d", &input);

    }

    return first;

}

*void* sort(*struct* Node \*\**ptr*) {

*struct* Node \*curr = \**ptr*, \*nex = NULL;

*int* temp;

    while(curr != NULL){

        nex = curr->next;

        while(nex != NULL){

            if(curr->data > nex->data){

                temp = curr->data;

                curr->data = nex->data;

                nex->data = temp;

            }

            nex = nex->next;

        }

        curr = curr->next;

    }

}

*void* display(*struct* Node \**ptr*){

    printf("Linked List: ");

    while(*ptr*!=0){

        printf("%d ", *ptr*->data);

*ptr* = *ptr*->next;

    }

    printf("\n");

}

*void* main(){

*struct* Node \*first = create();

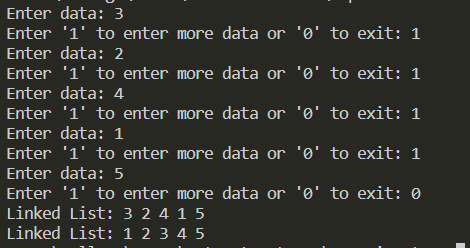
    display(first);

    sort(&first);

    display(first);

}

Output-



Ans 5(b)-

#include <stdio.h>

#include <stdlib.h>

*struct* node

{

*int* data;

*struct* node \*next;

} \*first = NULL, \*last, \*second = NULL;

*void* insertion(*int* *a*, *struct* node \**p*)

{

    static *int* flag = 0;

*struct* node \*t;

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

    t->data = *a*;

    t->next = NULL;

    if (*p* == 0)

    {

        last = t;

        if (flag == 0)

            first = t;

        else

            second = t;

    }

    else

    {

        last->next = t;

        last = t;

    }

    flag++;

}

*struct* node \*logic(*struct* node \**p*, *struct* node \**r*)

{

    while (*p* != 0)

    {

        if (*p*->data % 2 == 0 && *r* == NULL)

        {

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

*r*->data = *p*->data;

*r*->next = NULL;

            first = *r*;

            last = *r*;

*p* = *p*->next;

        }

        else if (*p*->data % 2 == 0 && *r* != NULL)

        {

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

*r*->data = *p*->data;

*r*->next = NULL;

            last->next = *r*;

            last = *r*;

*p* = *p*->next;

        }

        else

*p* = *p*->next;

    }

    return (*r*);

}

*struct* node \*concating()

{

*struct* node \*p, \*q, \*r;

    p = first;

    q = second;

    r = NULL;

    r = logic(p, r);

    logic(q, r);

    return (r);

}

*void* display(*struct* node \**p*)

{

    while (*p* != 0)

    {

        printf("%d ", *p*->data);

*p* = *p*->next;

    }

}

*int* main()

{

*struct* node \*p;

*int* n, a, x;

    printf("enter the number of elements in linked list a=");

    scanf("%d", &n);

    printf("enter the elements of list=\n");

    for (x = 0; x < n; x++)

    {

        scanf("%d", &a);

        insertion(a, first);

    }

    printf("enter the number of elements in linked list b=");

    scanf("%d", &n);

    printf("enter the elements of another list=\n");

    for (x = 0; x < n; x++)

    {

        scanf("%d", &a);

        insertion(a, second);

    }

    printf("current list a=");

    p = first;

    display(p);

    printf("\ncurrent list b=");

    p = second;

    display(p);

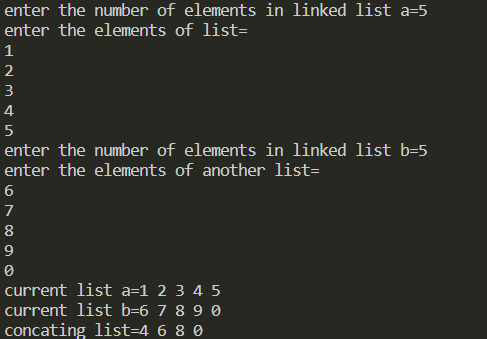
    p = concating();

    printf("\nconcating list=");

    display(p);

}

Output-



Ans 5(c)-

#include <stdio.h>

#include <stdlib.h>

*struct* node

{

*int* data;

*struct* node \*next;

} \*first = NULL, \*last;

*void* insertion(*int* *a*)

{

*struct* node \*t;

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

    t->data = *a*;

    t->next = NULL;

    if (first == NULL)

    {

        first = t;

        last = t;

    }

    else

    {

        last->next = t;

        last = t;

    }

}

*int* occurance(*int* *a*)

{

*struct* node \*p = first;

*int* sum = 0;

    while (p != 0)

    {

        if (p->data == *a*)

            sum++;

        p = p->next;

    }

    return (sum);

}

*void* display()

{

*struct* node \*p;

    p = first;

    while (p != 0)

    {

        printf("%d ", p->data);

        p = p->next;

    }

}

*int* main()

{

*int* n, a, x;

    printf("enter the number of elements in linked list=");

    scanf("%d", &n);

*int* arr[n];

    printf("enter the elements of linked list=\n");

    for (x = 0; x < n; x++)

    {

        scanf("%d", &a);

        insertion(a);

    }

    printf("current list= ");

    display();

*struct* node \*p = first;

    for (x = 0; x < n; x++)

    {

        arr[x] = occurance(p->data);

        p = p->next;

    }

    p = first;

    for (x = 0; x < n; x++)

    {

        printf("\noccurance of %d in LL= %d", p->data, arr[x]);

        while (p->data == p->next->data)

        {

            p = p->next;

            x++;

        }

        p = p->next;

    }

}

Output-

