ASSIGNMENT DAY-3

Name-Srujan

Question 1

Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume

position starts at 0.

Sol:-

void insertany(Node \*head ,int x)

{

Node \*temp,\*newnode,\*temp1;

int p;

newnode=(Node\*)malloc(sizeof(Node));

scanf("%d",&p);

newnode->data=p;

newnode->next=null;

if(x==0)

{

newnode->next=head;

newnode=head;

}

else{

for(int i=0;i<x;i++)

{

temp=temp1;

temp1=temp1->next;

}

temp->next=newnode;

newnode->next=temp1;

}

Question 2

Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.

Sol:-

void del\_beg(Node \*head)

{

Node \*temp;

temp=head;

head=head->next;

temp->next=null

}

Question 3

Write a function “delete\_end()” for deleting a node from the end of the linked list.

Void delete\_end(Node \*head)

{

Node \*temp,\*p;

temp=head;

while(temp->next!=null)

{

p=temp;

temp=temp->next;

}

p->next=null;

}

Question 4

In the Binary Search algorithm, it is suggested to calculate the mid as beg + (end - beg) / 2

instead of (beg + end) / 2. Why is it so?

Sol-

#include <stdio.h>

int binarySearch( int arr[], int item, int beg, int end) {

while (beg <= end) {

int midIndex = beg + (end - beg) / 2;

if (arr[midIndex] == item)

return midIndex;

if (arr[midIndex] > item)

beg = midIndex + 1;

else

end = midIndex - 1;

}

return -1;

}

int main(void) {

int arr[] = {13, 45, 65, 16, 117, 82, 19};

int n = sizeof(arr) / sizeof(arr[0]);

int item = 45;

int ans = binarySearch(arr, item, 0, n - 1);

if (ans == -1)

printf("Element not present");

else

printf("answer: %d", ans);

return 0;

}

To reduce the time complexity.

Question 5

Write the algorithm/function for Ternary Search.

Sol:-

double ternary\_search(double l, double r) {

double eps = 1e-9;

while (r - l > eps) {

double m1 = l + (r - l) / 3;

double m2 = r - (r - l) / 3;

double f1 = f(m1);

double f2 = f(m2);

if (f1 < f2)

l = m1;

else

r = m2;

}

return f(l);

}