Rajalakshmi Engineering College

Name: Naveen Prem
Email: 240801218@rajalakshmi.edu.in
Roll no: 240801218
Phone: 9443564728
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_PAH_modified

Attempt : 1
Total Mark : 5
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Imagine you are managing the backend of an e-commerce platform. Customers place orders at different times, and the orders are stored in two separate linked lists. The first list holds the orders from morning, and the second list holds the orders from the evening.

Your task is to merge the two lists so that the final list holds all orders in sequence from the morning list followed by the evening orders, in the same order

Input Format

The first line contains an integer n , representing the number of orders in the morning list.

The second line contains n space-separated integers representing the morning orders.

The third line contains an integer m , representing the number of orders in the evening list.

The fourth line contains m space-separated integers representing the evening orders.

Output Format

The output should be a single line containing space-separated integers representing the merged order list, with morning orders followed by evening orders.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
101 102 103
2
104 105
Output: 101 102 103 104 105

Answer

-

Status : Skipped ***Marks : 0/1***

2. Problem Statement

John is working on evaluating polynomials for his math project. He needs to compute the value of a polynomial at a specific point using a singly linked list representation.

Help John by writing a program that takes a polynomial and a value of x as input, and then outputs the computed value of the polynomial.

Example

Input:

2

13

12

11

1

Output:

36

Explanation:

The degree of the polynomial is 2.

Calculate the value of x2: 13 * 12 = 13.

Calculate the value of x1: 12 * 11 = 12.

Calculate the value of x0: 11 * 10 = 11.

Add the values of x2, x1 and x0 together: 13 + 12 + 11 = 36.

Input Format

The first line of input consists of the degree of the polynomial.

The second line consists of the coefficient x2.

The third line consists of the coefficient of x1.

The fourth line consists of the coefficient x0.

The fifth line consists of the value of x, at which the polynomial should be evaluated.

Output Format

The output is the integer value obtained by evaluating the polynomial at the given value of x.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
13
12
11
1
Output: 36

Answer

-

Status : Skipped ***Marks : 0/1***

3. Problem Statement

Bharath is very good at numbers. As he is piled up with many works, he decides to develop programs for a few concepts to simplify his work. As a first step, he tries to arrange even and odd numbers using a linked list. He stores his values in a singly-linked list.

Now he has to write a program such that all the even numbers appear before the odd numbers. Finally, the list is printed in such a way that all even numbers come before odd numbers. Additionally, the even numbers should be in reverse order, while the odd numbers should maintain their original order.

Example

Input:

6

3 1 0 4 30 12

Output:

12 30 4 0 3 1

Explanation:

Even elements: 0 4 30 12

Reversed Even elements: 12 30 4 0

Odd elements: 3 1

So the final list becomes: 12 30 4 0 3 1

Input Format

The first line consists of an integer n representing the size of the linked list.

The second line consists of n integers representing the elements separated by space.

Output Format

The output prints the rearranged list separated by a space.

The list is printed in such a way that all even numbers come before odd numbers and the even numbers should be in reverse order, while the odd numbers should maintain their original order.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 6
3 1 0 4 30 12
Output: 12 30 4 0 3 1

Answer

-

Status : Skipped ***Marks : 0/1***

4. Problem Statement

Write a program to manage a singly linked list. The program should allow

users to perform various operations on the linked list, such as inserting elements at the beginning or end, deleting elements from the beginning or end, inserting before or after a specific value, and deleting elements before or after a specific value. After each operation, the updated linked list should be displayed.

Input Format

The first line contains an integer choice, representing the operation to perform:

- For choice 1 to create the linked list. The next lines contain space-separated integers, with -1 indicating the end of input.
- For choice 2 to display the linked list.
- For choice 3 to insert a node at the beginning. The next line contains an integer data representing the value to insert.
- For choice 4 to insert a node at the end. The next line contains an integer data representing the value to insert.
- For choice 5 to insert a node before a specific value. The next line contains two integers: value (existing node value) and data (value to insert).
- For choice 6 to insert a node after a specific value. The next line contains two integers: value (existing node value) and data (value to insert).
- For choice 7 to delete a node from the beginning.
- For choice 8 to delete a node from the end.
- For choice 9 to delete a node before a specific value. The next line contains an integer value representing the node before which deletion occurs.
- For choice 10 to delete a node after a specific value. The next line contains an integer value representing the node after which deletion occurs.
- For choice 11 to exit the program.

Output Format

For choice 1, print "LINKED LIST CREATED".

For choice 2, print the linked list as space-separated integers on a single line. If the list is empty, print "The list is empty".

For choice 3, 4, 5, and 6, print the updated linked list with a message indicating the insertion operation.

For choice 7, 8, 9, and 10, print the updated linked list with a message indicating the deletion operation.

For any operation that is not possible print an appropriate error message such as

"Value not found in the list".

For choice 11 terminate the program.

For any invalid option, print "Invalid option! Please try again".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
5
3
7
-1
2
11
Output: LINKED LIST CREATED
5 3 7

Answer

-

Status : -

Marks : 0/1

5. Problem Statement

Emily is developing a program to manage a singly linked list. The program should allow users to perform various operations on the linked list, such as inserting elements at the beginning or end, deleting elements from the beginning or end, inserting before or after a specific value, and deleting elements before or after a specific value. After each operation, the updated linked list should be displayed.

Your task is to help Emily in implementing the same.

Input Format

The first line contains an integer choice, representing the operation to perform:

- For choice 1 to create the linked list. The next lines contain space-separated integers, with -1 indicating the end of input.
- For choice 2 to display the linked list.
- For choice 3 to insert a node at the beginning. The next line contains an integer data representing the value to insert.
- For choice 4 to insert a node at the end. The next line contains an integer data representing the value to insert.
- For choice 5 to insert a node before a specific value. The next line contains two integers: value (existing node value) and data (value to insert).
- For choice 6 to insert a node after a specific value. The next line contains two integers: value (existing node value) and data (value to insert).
- For choice 7 to delete a node from the beginning.
- For choice 8 to delete a node from the end.
- For choice 9 to delete a node before a specific value. The next line contains an integer value representing the node before which deletion occurs.
- For choice 10 to delete a node after a specific value. The next line contains an integer value representing the node after which deletion occurs.
- For choice 11 to exit the program.

Output Format

For choice 1, print "LINKED LIST CREATED".

For choice 2, print the linked list as space-separated integers on a single line. If the list is empty, print "The list is empty".

For choice 3, 4, 5, and 6, print the updated linked list with a message indicating the insertion operation.

For choice 7, 8, 9, and 10, print the updated linked list with a message indicating the deletion operation.

For any operation that is not possible print an appropriate error message such as "Value not found in the list".

For choice 11 terminate the program.

For any invalid option, print "Invalid option! Please try again".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
5
3
7
-1
2
11

Output: LINKED LIST CREATED
5 3 7

Answer

-

Status : -

Marks : 0/1

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 4_PAH

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Amar is working on a project where he needs to implement a special type of queue that allows selective dequeuing based on a given multiple.He wants to efficiently manage a queue of integers such that only elements not divisible by a given multiple are retained in the queue after a selective dequeue operation.

Implement a program to assist Amar in managing his selective queue.

Example

Input:

5

10230450

Status: Correct **Marks:10/1**

2.Problem Statement

Guide Harish in developing a simple queue system for a customer service center.The customer service center can handle up to 25 customers at a time.The queue needs to support basic operations such as adding a customer to the queue,serving a customer (removing them from the queue),and displaying the current queue of customers.

Use an array for implementation.

Input Format

The first line of the input consists of an integer N,the number of customers arriving at the service center.

The second line consists of N space-separated integers,representing the customer IDs in the order they arrive.

Output Format

After serving the first customer in the queue,display the remaining customers in the queue.

If a dequeue operation is attempted on an empty queue,display "Underflow".

Jf the queue is empty,display "Queue is empty".

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
101102103104105
Output:102103104105

Answer

#include <stdio.h>

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
2345938725
4
Output:Enqueued:23
Enqueued:45
Enqueued:93
Enqueued:87
Enqueued:25
The 4th largest element:25

Answer

```
//You are using GCC
#include<stdio.h>
#include<stdlib.h>

void enqueue(int queuell,int *rear,int element){
   if  (*rear<10){
      queue[(*rear)++]=element;
      printf("Enqueued:%d\n",element);
   }
}

int compare(const void *a,const void *b){
   retum(*(int*)b-*(int*)a);


int    main(){
   int   n,k;
   scanf("%d",&n);
   int   queue[10],rear   =0;

   for(inti    =0;i<n;i++){
      int element;
      scanf("%d",&element);
      enqueue(queue,&rear,element);
```

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
2345938725
4
Output:Enqueued:23
Enqueued:45
Enqueued:93
Enqueued:87
Enqueued:25
The 4th largest element:25

Answer

```
//You are using GCC
#include<stdio.h>
#include<stdlib.h>

void enqueue(int queuell,int *rear,int element){
   if  (*rear<10){
      queue[(*rear)++]=element;
      printf("Enqueued:%d\n",element);
   }
}

int compare(const void *a,const void *b){
   retum(*(int*)b-*(int*)a);


int    main(){
   int   n,k;
   scanf("%d",&n);
   int   queue[10],rear   =0;

   for(inti    =0;i<n;i++){
      int element;
      scanf("%d",&element);
      enqueue(queue,&rear,element);
```

The second line consists of a single integer,representing the unique identifier of each submitted ticket,separated by a space.

Output Format

The first line displays the "Queue:"followed by the ticket identifiers in the queue after all tickets have been submitted.

The second line displays the "Queue After Dequeue:"followed by the ticket identifiers in the queue after processing(removing)the ticket at the front.

Refer to the sample output for the exact text and format.

Sample Test Case

Input:6
145263956849
Output:Queue:145263956849
Queue After Dequeue:5263956849

Answer

```
//You are using GCC
#include<stdio.h>

int  main){
    int n;
    scanf("%d",&n);
    int queue[20];

    for(inti       =0;i<n;i++){
       scanf("%d",&queue[i]);


    printf("Queue:");
    for(inti    =0;i<n;i++){
       printf("%d",queueil);
    )
    printf("\n");

    printf("Queue After Dequeue:");
    for(inti    =1;i<n;i++){
```

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
2345938725
4
Output:Enqueued:23
Enqueued:45
Enqueued:93
Enqueued:87
Enqueued:25
The 4th largest element:25

Answer

```
//You are using GCC
#include<stdio.h>
#include<stdlib.h>

void enqueue(int queuell,int *rear,int element){
   if  (*rear<10){
      queue[(*rear)++]=element;
      printf("Enqueued:%d\n",element);
   }
}

int compare(const void *a,const void *b){
   retum(*(int*)b-*(int*)a);


int    main(){
   int  n,k;
   scanf("%d",&n);
   int   queue[10],rear   =0;

   for(inti      =0;i<n;i++){
      int element;
      scanf("%d",&element);
      enqueue(queue,&rear,element);
```

```
    scanf("%d",&customer_id);
    enqueue(customer_id);


  dequeue();


  displayQueue();

  retum 0;
```

Status:Correct

Marks:10/10

3.Problem Statement

Sharon is developing a queue using an array.She wants to provide the functionality to find the Kth largest element.The queue should support the **addition and retrieval of the Kth largest element effectively.The maximum** capacity of the queue is 10.

Assist her in the program.

Input Format

The first line of input consists of an integer N,representing the number of elements in the queue.

The second line consists of N space-separated integers.

The third line consists of an integer K.

Output Format

For each enqueued element,print a message:"Enqueued:"followed by the element

The last line prints "The [K]th largest element:"followed by the Kth largest **element**

The second line consists of a single integer,representing the unique identifier of each submitted ticket,separated by a space.

Output Format

The first line displays the "Queue:"followed by the ticket identifiers in the queue after all tickets have been submitted.

The second line displays the "Queue After Dequeue:"followed by the ticket identifiers in the queue after processing(removing)the ticket at the front.

Refer to the sample output for the exact text and format.

Sample Test Case

Input:6
145263956849
Output:Queue:145263956849
Queue After Dequeue:5263956849

Answer

```
//You are using GCC
#include<stdio.h>

int  main){
    int n;
    scanf("%d",&n);
    int queue[20];

    for(inti       =0;i<n;i++){
       scanf("%d",&queue[i]);


    printf("Queue:");
    for(inti    =0;i<n;i++){
       printf("%d",queueil);
    )
    printf("\n");

    printf("Queue After Dequeue:");
    for(inti    =1;i<n;i++){
```

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify results

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 6_PAH_Updated

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Vishnu,a math enthusiast,is given a task to explore the magic of numbers He has an array of positive integers,and his goal is to find the integer with the highest digit sum in the sorted array using the merge sort algorithm.

You have to assist Vishnu in implementing the merge sort algorithm.

Input Format

The first line of input consists of an integer N,representing the number of **elements in the array.**

The second line consists of N space-separated integers,representing the array **elements.**

Output Format

```
    R[i]=arr[mid+1+jl;
    i=0;
    j=0;
    k=left;

    //Merge the temp arrays back
    while(i<n1      &&j<n2){
       if  (L[i]<=R[jJ)
          arr[k++]=L[i++];
       else
          arr[k++]=R[j++];
    }

    //Copy remaining elements
    while(i<n1)
       arr[k++]=L[i++];
    while   (j<n2)
       arr[k++]=R[j++];


//Merge  sort  function
void  mergeSort(int  arrl,int  left,int  right){
  if    (left<right){
     int        mid=left+(right-left)/2;

     mergeSort(arr,left,mid);
     mergeSort(arr,mid+1,right);

     merge(arr,left,mid,right);


int   main){
  int  N;
  scanf("%d",&N);

  int       ar[10];//since1≤N≤10
  for(inti =0;i<N;i++)
     scanf("%d",&arr[i]);

  //Sort the aray using merge sort
```

```
    R[i]=arr[mid+1+jl;
    i=0;
    j=0;
    k=left;

    //Merge the temp arrays back
    while(i<n1      &&j<n2){
       if  (L[i]<=R[jJ)
          arr[k++]=L[i++];
       else
          arr[k++]=R[j++];
    }

    //Copy remaining elements
    while(i<n1)
       arr[k++]=L[i++];
    while   (j<n2)
       arr[k++]=R[j++];


//Merge  sort  function
void  mergeSort(int  arrl,int  left,int  right){
  if    (left<right){
     int        mid=left+(right-left)/2;

     mergeSort(arr,left,mid);
     mergeSort(arr,mid+1,right);

     merge(arr,left,mid,right);


int   main){
  int  N;
  scanf("%d",&N);

  int       ar[10];//since1≤N≤10
  for(inti =0;i<N;i++)
     scanf("%d",&arr[i]);

  //Sort the aray using merge sort
```

Output Format

The output prints the sorted finishing times of the athletes in ascending order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
7589659070
Output:6570758990

Answer

```
/You are using GCC
#include <stdio.h>

void insertionSort(int arr[,int n){
   for(inti=1;i<n;i++){
      int key =arr[i];
      intj=i-1;

      //Shift elements that are greater than key one position ahead
      while(j>=0     &&arr[i]>key){
         arr[j+1]=arr[j;
        j-;

      arr[j+1]=key;


int   main){
   int n;
   scanf("%d",&n);//Read number of athletes

   int times[n];
   for(inti      =0;i<n;i++){
      scanf("%d",&times[i]);//Read finishing times


   insertionSort(times,n);
```

```
//Print sorted finishing
times for(inti=0;i<n;i++)
    printf("%d",times[i);

printf("\n");

retum 0
```

Status:Correct

Marks:10/10

3. Problem Statement

You are working as a programmer at a sports academy,and the academy holds various sports competitions regularly.

As part of the academy's system,you need to sort the scores of the participants in descending order using the Quick Sort algorithm.

Write a program that takes the scores of n participants as input and uses the Quick Sort algorithm to sort the scores in descending order.Your program should display the sorted scores after the sorting process.

Input Format

The first line of input consists of an integer n,which represents the number of scores.

The second line of input consists of n integers,which represent scores separated by spaces.

Output Format

Each line of output represents an iteration of the Quick Sort algorithm,displaying **the elements of the array at that iteration.**

After the iterations are complete,the last line of output prints the sorted scores in descending order separated by space.

```
//Print sorted finishing
times for(inti=0;i<n;i++)
    printf("%d",times[i);

printf("\n");

retum 0
```

Status:Correct

Marks:10/10

3. Problem Statement

You are working as a programmer at a sports academy,and the academy holds various sports competitions regularly.

As part of the academy's system,you need to sort the scores of the participants in descending order using the Quick Sort algorithm.

Write a program that takes the scores of n participants as input and uses the Quick Sort algorithm to sort the scores in descending order.Your program should display the sorted scores after the sorting process.

Input Format

The first line of input consists of an integer n,which represents the number of scores.

The second line of input consists of n integers,which represent scores separated by spaces.

Output Format

Each line of output represents an iteration of the Quick Sort algorithm,displaying **the elements of the array at that iteration.**

After the iterations are complete,the last line of output prints the sorted scores in descending order separated by space.

```
//Print sorted finishing
times for(inti=0;i<n;i++)
    printf("%d",times[i);

printf("\n");

retum 0
```

Status:Correct

Marks:10/10

3. Problem Statement

You are working as a programmer at a sports academy,and the academy holds various sports competitions regularly.

As part of the academy's system,you need to sort the scores of the participants in descending order using the Quick Sort algorithm.

Write a program that takes the scores of n participants as input and uses the Quick Sort algorithm to sort the scores in descending order.Your program should display the sorted scores after the sorting process.

Input Format

The first line of input consists of an integer n,which represents the number of scores.

The second line of input consists of n integers,which represent scores separated by spaces.

Output Format

Each line of output represents an iteration of the Quick Sort algorithm,displaying **the elements of the array at that iteration.**

After the iterations are complete,the last line of output prints the sorted scores in descending order separated by space.

```
//Print sorted finishing
times for(inti=0;i<n;i++)
    printf("%d",times[i);

printf("\n");

retum 0
```

Status:Correct

Marks:10/10

3. Problem Statement

You are working as a programmer at a sports academy,and the academy holds various sports competitions regularly.

As part of the academy's system,you need to sort the scores of the participants in descending order using the Quick Sort algorithm.

Write a program that takes the scores of n participants as input and uses the Quick Sort algorithm to sort the scores in descending order.Your program should display the sorted scores after the sorting process.

Input Format

The first line of input consists of an integer n,which represents the number of scores.

The second line of input consists of n integers,which represent scores separated by spaces.

Output Format

Each line of output represents an iteration of the Quick Sort algorithm,displaying **the elements of the array at that iteration.**

After the iterations are complete,the last line of output prints the sorted scores in descending order separated by space.

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 5_PAH_Updated

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Joseph,a computer science student,is interested in understanding binary search trees (BST)and their node arrangements.He wants to create a program to explore BSTs by inserting elements into a tree and displaying the nodes using post-order traversal of the tree.

Write a program to help Joseph implement the program.

Input Format

The first line of input consists of an integer N,representing the number of elements to insert into the BST.

The second line consists of N space-separated integers data,which is the data to be inserted into the BST.

Output Format

The output prints N space-separated integer values after the post-order traversal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input:4
101553
Output:351510

Answer

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

struct Node{
    int data;
    struct Node*left;
    struct Node*right;
};

struct Node*createNode(int data){
    struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->data =data;
    newNode->left =newNode->right =NULL;
    return newNode;
}

struct Node*insert(struct Node*root,int data){
    if (root ==NULL){
        return createNode(data);

    if (data<root->data){
        root>left =insert(root->left,data);
    }else {
        root->right =insert(root->right,data);

    return root;
```

```
void postorder(struct Node*root){
   if (root !=NULL){
      postorder(root>left);
      postorder(root->right);
      printf("%d",root->data);
   }
}

int  main){
   struct Node*root =NULL;
   int N,data;

   scanf("%d",&N);
   for(inti=0;i<N;i++){
      scanf("%d",&data);
      root =insert(root,data);


   postorder(root);
   printf("\n");

   return 0;
```

Status:Correct

Marks:10/10

2.Problem Statement

Yogi is working on a program to manage a binary search tree (BST) containing integer values.He wants to implement a function that removes nodes from the tree that fall outside a specified range defined by a
minimum and maximum value.

Help Yogi by writing a function that achieves this.

Input Format

The first line of input consists of an integer N,representing the number of elements to be inserted into the BST.

The second line consists of N space-separated integers,representing the elements to be inserted into the BST.

The third line consists of two space-separated integers min and max, representing the minimum value and the maximum value of the range.

Output Format

The output prints the remaining elements of the BST in an in-order traversal, after removing nodes that fall outside the specified range.

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5

105152012

515

Output:5101215

Answer

```
//You are using GCC
#include <stdio.h>
#include <stdlib.h>

struct Node{
  int data;
  struct Node*left;
  struct Node*right;
};

struct Node*createNode(int data){
  struct Node*newNode =(struct Node*)malloc(sizeof(struct Node));
  newNode->data =data;
  newNode->left =newNode->right =NULL;
  return newNode;


struct Node*insert(struct Node*root,int data){
  if (root ==NULL){
```

```
    return createNode(data);

  if (data<root->data){
    root->left =insert(root->left,data);
  }else {
    root->right =insert(root->right,data);

  return root;


struct Node*trimBST(struct Node*root,int min,int max){
  if (root ==NULL){
    return NULL;
  }n
  root->left =trimBST(root->left,min,max);
  root->right =trimBST(root->right,min,max);

  if (root->data<min){
    struct Node*rightChild =root->right;
    free(root);
    return rightChild;
  }
  if (root->data>max){
    struct Node*leftChild =root->left;
    free(root);
    return leftChild;

  return root;


void inorder(struct Node*root){
  if (root !=NULL){
    inorder(root->left);
    printf("%d",root->data);
    inorder(root->right);

}

int main){
  struct Node*root =NULL;
  int N,data,min,max;
```

```
scanf("%d",&N);
for(inti=0;i<N;i++){
   scanf("%d",&data);
   root   =insert(root,data);
  }
scanf("%d %d",&min,&max);
root   =trimBST(root,min,max);

inorder(root);
printf('in";

   return 0;
```

Marks:10/10

Status:*Correct*

3.Problem Statement

Arun is exploring operations on binary search trees(BST).He wants to write a program with an unsorted distinct integer array that represents the BST keys and construct a height-balanced BST from it.

After constructing,he wants to perform the following operations that can alter the structure of the tree and traverse them using a level-order

traversal:

InsertionDeletion

Your task is to assist Arun in completing the program without any errors.

Input Format

The first line of input consists of an integer N,representing the number of initial keys in the BST.

The second line consists of N space-separated integers,representing the initial keys.

The third line consists of an integer X,representing the new key to be inserted sinto the BST.

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 2_PAH

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Bala is a student learning about the doubly linked list and its functionalities.He came across a problem where he wanted to create a doubly linked list by appending elements to the front of the list.

After populating the list,he wanted to delete the node at the given position from the beginning.Write a suitable code to help Bala.

Input Format

The first line contains an integer N,the number of elements in the doubly linked list.

The second line contains N integers separated by a space,the data values of the nodes in the doubly linked list.

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 2_PAH

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Bala is a student learning about the doubly linked list and its functionalities.He came across a problem where he wanted to create a doubly linked list by appending elements to the front of the list.

After populating the list,he wanted to delete the node at the given position from the beginning.Write a suitable code to help Bala.

Input Format

The first line contains an integer N,the number of elements in the doubly linked list.

The second line contains N integers separated by a space,the data values of the nodes in the doubly linked list.

The third line contains an integer X,the position of the node to be deleted from the doubly linked list.

Output Format

The first line of output displays the original elements of the doubly linked list, separated by a space.

The second line prints the updated list after deleting the node at the given position X from the beginning.

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5

1020304050

2

Output:5040302010

50302010

Answer

```
//You are using GCC
#include <stdio.h>
#include<stdlib.h>

  Define the structure of a doubly linked list node
struct  Node{
  int data;
  struct  Node*prev;
  struct  Node*next;
};

//Function to insert a node at the front
void insertFront(struct  Node**head_ref,int  data){
  struct  Node*newNode  =(struct  Node*)malloc(sizeof(struct
  Node)); newNode->data  =data;
  newNode->prev  =NULL;
  newNode->next  =*head_ref;

  if  (*head_ref !=NULL)
```

```
    (*head_ref)->prev =newNode;
  *head_ref =newNode;

//Function to delete a node at position X
void deleteAtPosition(struct Node**head_ref,int pos){
  if (*head_ref ==NULLll pos <=0)
    return;

  struct Node*temp =*head_ref;

  //Traverse to the position
  for(int i=1;temp !=NULL &&i<pos;i++)
    temp =temp->next;

  if (temp ==NULL)return;

  if (temp->prev !=NULL)
    temp->prev->next =temp->next;
  else
    *head_ref =temp->next;//if head is being removed

  if (temp->next !=NULL)
    temp->next->prev =temp->prev;

  free(temp);


//Function to print the list
void printList(struct Node*head){
  while(head !=NULL){
    printf("%d",head->data);
    head =head->next;

  printf("\n");
}

int main(){
  int N,X,value;
  struct Node*head =NULL;
```

```
    (*head_ref)->prev =newNode;
  *head_ref =newNode;

//Function to delete a node at position X
void deleteAtPosition(struct Node**head_ref,int pos){
  if (*head_ref ==NULL|| pos <=0)
    return;

  struct Node*temp =*head_ref;

  //Traverse to the position
  for(int i=1;temp !=NULL &&i<pos;i++)
    temp =temp->next;

  if (temp ==NULL)return;

  if (temp->prev !=NULL)
    temp->prev->next =temp->next;
  else
    *head_ref =temp->next;//if head is being removed

  if (temp->next !=NULL)
    temp->next->prev =temp->prev;

  free(temp);


//Function to print the list
void printList(struct Node*head){
  while(head !=NULL){
    printf("%d",head->data);
    head =head->next;

  printf("\n");
}

int main(){
  int N,X,value;
  struct Node*head =NULL;
```

Rajalakshmi Engineering College

Name:Naveen Prem
Email:240801218@rajalakshmi.edu.in
Roll no:240801218
Phone:9443564728
Branch:REC
Department:I ECE AF
Batch:2028
Degree:B.E-ECE

Scan to verify

NeoColab REC CS23231 DATA STRUCTURES

REC_DS using C_Week 4_PAH

Attempt:1
Total Mark:50
Marks Obtained:50

Section 1:Coding

1.Problem Statement

Amar is working on a project where he needs to implement a special type of queue that allows selective dequeuing based on a given multiple.He wants to efficiently manage a queue of integers such that only elements not divisible by a given multiple are retained in the queue after a selective dequeue operation.

Implement a program to assist Amar in managing his selective queue.

Example

Input:

5

10230450

5

Output:

Original Queue:10230450

Queue after selective dequeue:24

Explanation:

After selective dequeue with a multiple of 5,the elements that are multiples of 5 should be removed.Therefore,only 10,30,and 50 should be removed from the queue.The updated Queue is 24.

Input Format

The first line contains an integer n,representing the number of elements initially present in the queue.

The second line contains n space-separated integers,representing the elements of the queue

The third line contains an integer multiple,representing the divisor for selective dequeue operation.

Output Format

The first line of output prints "Original Queue:"followed by the space-separated elements in the queue before the dequeue operation.

The second line prints "Queue after selective dequeue:"followed by the remaining space-separated elements in the queue,after deleting elements that are the multiples of the specified number.

Refer to the sample output for the formatting specifications.

Sample Test Case

nput:5
10230450
5
Output Original Queue:10230450
Queue after selective dequeue:24

rotate the list.

Output Format

The output displays the elements of the doubly linked list after rotating it by k positions.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input:5
12345

Output:51234

Answer

```
//You are using GCC
#include <stdio.h>
#include <stdlib.h>

//Node structure
struct Node{
    int data;
    struct Node*prev;
    struct Node*next;


//Function to create a new node
struct Node*createNode(int data){
    struct Node*newNode =(struct Node*)malloc(sizeof(struct Node));
    newNode->data =data;
    newNode>prev =newNode->next =NULL;
    return newNode;
  }

//Function to insert a node at the end
void insertEnd(struct Node**head_ref,int data){
    struct Node*newNode =createNode(data);

    if (*head_ref ==NULL){
```

Status: Correct

Marks:10/1

2.Problem Statement

Guide Harish in developing a simple queue system for a customer service center.The customer service center can handle up to 25 customers at a time.The queue needs to support basic operations such as adding a customer to the queue,serving a customer (removing them from the queue),and displaying the current queue of customers.

Use an array for implementation.

Input Format

The first line of the input consists of an integer N,the number of customers arriving at the service center.

The second line consists of N space-separated integers,representing the customer IDs in the order they arrive.

Output Format

After serving the first customer in the queue,display the remaining customers in the queue.

If a dequeue operation is attempted on an empty queue,display "Underflow".

Jf the queue is empty,display "Queue is empty".

Refer to the sample output for formatting specifications.

Sample Test Case

Input:5
101102103104105
Output:102103104105

Answer

#include <stdio.h>