Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 0_Arrays and Functions

Attempt : 3
Total Mark : 5
Marks Obtained : 5

Section 1 : Coding

1. Problem Statement

Alex, a budding programmer, is tasked with writing a menu-driven program to perform operations on an array of integers. The operations include finding the smallest number, the largest number, the sum of all numbers, and their average. The program must repeatedly display the menu until Alex chooses to exit.

Write a program to ensure the specified tasks are implemented based on Alex's choices.

Input Format

The first line contains an integer n, representing the number of elements in the array.

The second line contains n space-separated integers representing the array elements.

The subsequent lines contain integers representing the menu choices:

Choice 1: Find and display the smallest number.

Choice 2: Find and display the largest number.

Choice 3: Calculate and display the sum of all numbers.

Choice 4: Calculate and display the average of all numbers as double.

Choice 5: Exit the program.

Output Format

For each valid menu choice, print the corresponding result:

For choice 1, print "The smallest number is: X", where X is the smallest number in the array.

For choice 2, print "The largest number is: X", where X is the largest number in the array.

For choice 3, print "The sum of the numbers is: X", where X is the sum of all numbers in the array.

For choice 4, print "The average of the numbers is: X. XX", where X.XX is the double value representing an average of all numbers in the array, rounded to two decimal places.

For choice 5, print "Exiting the program".

If an invalid choice is made, print "Invalid choice! Please enter a valid option (1-5)."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
10 20 30
1
5

Output: The smallest number is: 10
Exiting the program

Answer

```
// You are using GCC
#include<stdio.h>
int main(){
    int n,small=0,large=0,sum=0;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
        sum+=arr[i];
    }
    small=arr[0];
    large=arr[0];
    float avg=((sum*1.0)/n);
    for(int i=0;i<n;i++){
        if(arr[i]<small){
            small=arr[i];
        }
        else if(arr[i]>large){
            large=arr[i];
        }
    }
    while(1){
        int a;
        scanf("%d",&a);
        if(a==1){
            printf("The smallest number is: %d\n",small);
        }
        else if(a==2){
            printf("The  largest  number  is:  %d\n",large);
        }
        else if(a==3){
            printf("The sum of the numbers is : %d\n",sum);
        }
        else if (a==4){
            printf("The average of the numbers is: %.2f\n",avg);
        }
        else if(a==5){
            printf("Exiting the program");
            break;
        }
        else{
            printf("Invalid choice! Please enter a valid option (1-5).\n");
        }
    }
    return 0;
}
```

Status : Correct *Marks : 1/1*

2. Problem Statement

Tim is creating a program to track and analyze student attendance. The program requires two inputs: the total number of students (n) and the total number of class sessions (m). The task is to design and populate an attendance matrix, 'matrix', representing the attendance record of each student for each session.

The program's specific objective is to determine whether the last student on the list attended an even or odd number of classes. This functionality will aid teachers in quickly evaluating the attendance habits of individual students.

Input Format

The first line of input consists of a positive integer n, representing the number of students.

The second line consists of a positive integer m, representing the number of class sessions.

The next n lines consist of m space-separated positive integers representing the number of classes attended by the student.

Output Format

The output displays one of the following results:

If the last session is even the output prints "[LastSession] is even".

If the last session is odd the output prints "[LastSession] is odd".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 2
2
1 2
3 100

Output: 100 is even

Answer

```
// You are using GCC
#include<stdio.h>
int main(){
   int n,m;
   scanf("%d %d",&n,&m);
   int matrix[n][m];
   for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
         scanf("%d",&matrix[i][j]);
      }
   }
   if(matrix[n-1][m-1]%2==0){
      printf("%d is even",matrix[n-1][m-1]);
   }
   else{
      printf("%d is odd",matrix[n-1][m-1]);
   }
   return 0;
}
```

Status : Correct *Marks : 1/1*

3. Problem Statement

Saurabh is the manager of a growing tech company. He needs a program to record and analyze the monthly salaries of his employees. The program will take the number of employees and their respective salaries as input and then calculate the average salary, and find the highest and lowest salary among them.

Help Saurabh automate this task efficiently.

Input Format

The first line of input consists of an integer n, representing the number of employees.

The second line consists of n integers, where each integer represents the salary of an employee.

Output Format

The output prints n lines, where each line will display: "Employee i: "Salary

Where i is the employee number (starting from 1) and salary is the respective salary of that employee.

After that, print the average salary in the following format: "Average Salary: "average_salary

Where average_salary is the average salary of all employees, rounded to two decimal places.

Next, print the highest salary in the following format: "Highest Salary: "max_salary

Where max_salary is the highest salary among all employees.

Finally, print the lowest salary in the following format:"Lowest Salary: "min_salary

Where min_salary is the lowest salary among all employees.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 5
4000
3500
6000
2500
4500
```

```
Output: Employee 1: 4000
Employee 2: 3500
Employee 3: 6000
Employee 4: 2500
Employee 5: 4500

Average Salary: 4100.00
Highest Salary: 6000
Lowest Salary: 2500
```

Answer

```
// You are using GCC
#include<stdio.h>
int main(){
   int n,high=0;
   float total=0.0;
   scanf("%d",&n);
   int arr[n];
   for(int i=0;i<n;i++){
      scanf("%d",&arr[i]);
      total+=arr[i]*1.0;
      if(arr[i]>high){
         high=arr[i];
      }
```

```
    }
    int low=arr[0];
    for(int i=1;i<=n;i++){
        printf("Employee %d: %d\n",i,arr[i-1]);
        if(arr[i-1]<low){
            low=arr[i-1];
        }
    }
    printf("\nAverage Salary: %.2f\n",total/n);
    printf("Highest Salary: %d\n",high);
    printf("Lowest Salary: %d\n",low);
    return 0;
}
```

Status : Correct *Marks : 1/1*

4. Problem Statement

Write a program that will read a Matrix (two-dimensional arrays) and print the sum of all elements of each row by passing the matrix to a function.

Function Signature: void calculateRowSum(int [ ][ ], int, int)

Input Format

The first line consists of an integer M representing the number of rows.

The second line consists of an integer N representing the number of columns.

The next M lines consist of N space-separated integers in each line representing the elements of the matrix.

Output Format

The output displays the sum of all elements of each row separated by a space.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
3
1 2 3
4 5 6
7 8 9

Output: 6 15 24

Answer

```
#include <stdio.h>

// You are using GCC
void calculateRowSum(int matrix[20][20], int rows, int cols) {
   //Type your code here
   int sum[rows];
   for(int i=0;i<rows;i++){
      sum[i]=0;
      for(int z=0;z<cols;z++){
         sum[i]+=matrix[i][z];
      }
   }
   for(int i=0;i<rows;i++){
      printf("%d ",sum[i]);
   }
}

int main() {
   int matrix[20][20];
   int r, c;

   scanf("%d", &r);
   scanf("%d", &c);

   for (int i = 0; i < r; i++) {
      for (int j = 0; j < c; j++) {
         scanf("%d", &matrix[i][j]);
      }
   }

   calculateRowSum(matrix, r, c);
   return 0;
}
```

Status : Correct *Marks : 1/1*

5. Problem Statement

Write a program that reads an integer 'n' and a square matrix of size 'n x n' from the user. The program should then set all the elements in the lower triangular part of the matrix (including the main diagonal) to zero using a function and display the resulting matrix.

Function Signature: void setZeros(int [ ][ ], int)

Input Format

The first line consists of an integer M representing the number of rows & columns.

The next M lines consist of M space-separated integers in each line representing the elements of the matrix.

Output Format

The output displays the matrix containing M space-separated elements in M lines where the lower triangular elements are replaced with zero.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 3
10 20 30
40 50 60
70 80 90
```

```
Output: 0 20 30
0 0 60
0 0 0
```

Answer

```
#include <stdio.h>

// You are using GCC
void setZeros(int arr[10][10], int n) {
   //Type your code here
   for(int i=0;i<n;i++){
       for(int j=0;j<=i;j++){
           arr[i][j]=0;
       }
   }
}

int main() {
   int arr1[10][10];
   int n;

   scanf("%d", &n);

   for (int i = 0; i < n; i++) {
       for (int j = 0; j < n; j++) {
           scanf("%d",  &arr1[i][j]);
       }
   }

   setZeros(arr1, n);

   for (int i = 0; i < n; i++) {
       for (int j = 0; j < n; j++) {
           printf("%d ", arr1[i][j]);
       }
       printf("\n");
   }

   return 0;
}
```

Status : Correct *Marks : 1/1*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 0_Pointers

Attempt : 1
Total Mark : 5
Marks Obtained : 5

Section 1 : Coding

1. Problem Statement

Daniel is working on a project that involves analyzing data stored in float arrays. He needs to determine whether a given float array contains only positive numbers.

To achieve this, he needs a program that can accurately evaluate the contents of float arrays using malloc().

Input Format

The first line of input consists of an integer N, representing the size of the array.

The second line consists of N space-separated float values, representing the elements of the array.

Output Format

If all the array elements are positive, print "All elements are positive."

If the array contains at least one positive element, print "At least one element is positive."

If there are no positive elements in the array, print "No positive elements in the array."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
50.0 -2.3 3.7 -4.8 5.2

Output: At least one element is positive.

Answer

```
#include <stdio.h>
#include   <stdlib.h>
int main(){
   int n,pos=0;
   scanf("%d",&n);
   float *arr;
   arr=(float*)malloc(n*sizeof(float));
   for(int i=0;i<n;i++){
      scanf("%f",&arr[i]);
      if(arr[i]>0){
         pos++;
      }
   }
   if(pos==n){
      printf("All elements are positive.");
   }
   else if(pos>0){
      printf("At least one element is positive.");
   }
   else{
      printf("No positive elements in the array.");
   }
   return 0;
}
```

Status : Correct *Marks : 1/1*

2. Problem Statement

Ria is a mathematician who loves exploring combinatorics. She is working on a project that involves calculating permutations.

Ria wants to create a program that takes the values of n and r as input and calculates the permutations of n elements taken r at a time.

Write a program using pointers and a function calculatePermutations that, given the values of n and r, calculates and prints the permutations of n elements taken r at a time.

Permutation: n! / (n - r)!

Input Format

The first line consists of an integer n, representing the total number of elements.

The second line consists of an integer r, representing the number of elements to be taken at a time.

Output Format

The output prints the result of the permutation.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
3

Output: 24

Answer

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

```
long long calculate(int n, int r){
   long long per=1;
   for(int i=0;i<r;i++){
      per*=(n-i);
   }
   return per;
}
int main(){
   int n,r;
   scanf("%d %d",&n,&r);
   if(n<0||r<0||r>24||n>25){
      return 1;
   }
   printf("%lld\n",calculate(n,r));
   return 0;
}
```

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

3. Problem Statement

Raj wants to create a program using pointers and a structure named Employee to manage employee information.

He seeks your assistance to input the employee's name, salary, and hours worked. Implement a salary increase based on hours worked, and calculate the final salary. Calculate the total salary for 30 days. Display the results of the final and total salary.

Salary increase criteria:

If hours worked >= 12, the increase is Rs. 150.00.If hours worked >= 10, but less than 12, the increase is Rs. 100.00.If hours worked >= 8, but less than 10, the increase is Rs. 50.00.If hours worked < 8, there is no increase.

Input Format

The first line of input consists of a string, representing the Employee's name.

The second line consists of a double-point number, representing the Employee's current salary.

The third line consists of an integer, representing the number of hours worked by the employee.

Output Format

The first line of output prints "Final Salary: Rs. " followed by a double value, representing the final salary, rounded off to two decimal places.

The second line prints "Total Salary: Rs. " followed by a double value, representing the total salary for 30 days, rounded off to two decimal places.

Refer to the sample outputs for formatting specifications.

Sample Test Case

Input: Akil
3000.00
6

Output: Final Salary: Rs. 3000.00
Total Salary: Rs. 90000.00

Answer

```
#include <stdio.h>
struct Employee{
   char name[50];
   float salary;
   int hr;
};
int main(){
   struct Employee e;
   scanf("%s %f %d",&e.name,&e.salary,&e.hr);
   if(e.hr>=12){
      e.salary+=150.0;
   }
   else if(e.hr>=10){
      e.salary+=100.0;
   }
   else if(e.hr>=8){
      e.salary+=50.0;
   }
```

```
    printf("Final Salary: Rs. %.2f",e.salary);
    printf("Total Salary: Rs. %.2f",e.salary*30);
    return 0;
}
```

Status : Correct *Marks : 1/1*

4. Problem Statement

Rajwinder wants a program to determine retirement details for a person based on their age.

Create a program that uses a structure called Person to hold the age as an attribute with a pointer.

If the age is under 18, display "Invalid".If the age is 65 or older, print "Already retired!".Otherwise, calculate and output the retirement year, remaining years, and remaining days until retirement.

Note: Age 65 is considered as retirement age. Assume the current year as 2023 and there are 365 days per year for calculation.

Input Format

The input consists of an integer representing the person's age.

Output Format

If the age is under 18, the output displays "Invalid" and terminates.

If the age is 65 or older, the output displays "Already retired!" and terminates.

Otherwise, the output displays the following.

1. The first line displays "Retirement Year: " followed by an integer representing the retirement year.
2. The second line displays "Remaining Years: " followed by an integer representing the remaining years left for retirement.
3. The third line displays "Remaining Days: " followed by an integer representing the remaining days left for retirement.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 43

```
Output: Retirement Year: 2045
Remaining Years: 22
Remaining Days: 8030
```

Answer

```
#include <stdio.h>
struct Person{
   int age;
};
int main(){
   struct Person p;
   scanf("%d",&p.age);
   int rem = 65-p.age,ret=rem+2023,days=rem*365;
   if(p.age<18){
      printf("Invalid");
   }
   else if(p.age>=65){
      printf("Already retired!");
   }
   else{
      printf("Retirement Year: %d",ret);
      printf("Remaining Years: %d",rem);
      printf("Remaining  Days:  %d",days);
   }
   return 0;
}
```

Status : Correct *Marks : 1/1*

5. Problem Statement

Sam is developing a program for analyzing daily temperature fluctuations. Users input the number of days, followed by daily temperature values whose memory is allocated using malloc.

The program calculates and displays the following:

The absolute temperature changes between consecutive days (The first value remains the same).The average temperature of adjacent days.

This allows users to gain insights into daily temperature variations for better analysis.

For Example,

Let us assume the temperature for 3 days as 25.5, 28.0, and 23.5.

The absolute differences:

Day 1: (N/A, as there is no previous day) = 25.50Day 2: abs(28.0 - 25.5) = 2.50Day 2: abs(23.5 - 28.0) = 4.50

The average temperatures:

Day 1: (N/A, as there is no previous day) = 25.50Day 2: (25.5 + 23.5) / 2.0 = 24.50Day 3: (N/A, as there is no next day) = 23.50

Input Format

The first line consists of an integer N, representing the number of days.

The second line consists of N space-separated float values, representing the temperature values for N days.

Output Format

The first line displays the absolute temperature change for N days as float values, rounded to two decimal places, separated by a space.

The second line displays the average temperature for N days as float values, rounded to two decimal places, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
25.5 28.0 23.5

```
Output:  25.50  2.50  4.50
25.50 24.50 23.50
```

Answer

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main(){
   int n;
   scanf("%d",&n);
   float *temp=(float*) malloc(n*sizeof(float));
   for(int i=0;i<n;i++)
   scanf("%f",&temp[i]);
   printf("%.2f ",temp[0]);
   for(int i=1;i<n;i++)
   printf("%.2f ",fabs(temp[i]-temp[i-1]));
   printf("\n");
   printf("%.2f ",temp[0]);
   for(int i=1;i<n-1;i++){
      printf("%.2f ",(temp[i-1]+temp[i+1])/2.0);
      temp[i]=(temp[i-1]+temp[i+1])/2.0;
   }
   printf("%.2f",temp[n-1]);
   free(temp);
   return 0;
}
```

Status : Correct *Marks : 1/1*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email:241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Janani is a tech enthusiast who loves working with polynomials. She wants to create a program that can add polynomial coefficients and provide the sum of their coefficients.

The polynomials will be represented as a linked list, where each node of the linked list contains a coefficient and an exponent. The polynomial is represented in the standard form with descending order of exponents.

Input Format

The first line of input consists of an integer n, representing the number of terms in the first polynomial.

The following n lines of input consist of two integers each: the coefficient and the exponent of the term in the first polynomial.

The next line of input consists of an integer m, representing the number of terms in the second polynomial.

The following m lines of input consist of two integers each: the coefficient and the exponent of the term in the second polynomial.

Output Format

The output prints the sum of the coefficients of the polynomials.

Sample Test Case

```
Input: 3
2 2
3 1
4 0
3
2 2
3 1
4 0
Output: 18
```

Answer

```
#include  <stdio.h>
#include  <stdlib.h>
struct node{
   int coff;
   int power;
   struct node* next;
};
void  insert(struct  node**  head,int  coff,int  power){
   struct node* newnode = (struct  node*)malloc(sizeof(struct node));
   newnode->coff = coff;
   newnode->power = power;
   newnode->next = NULL;
   if(*head ==NULL || (*head)->power<power){
      newnode->next = *head;
      *head=newnode;
   }
   else{
      struct  node*  temp=*head;
      while(temp->next!=NULL && temp->next->power>=power){
         temp = temp->next;
```

```
        }
        temp->next=newnode;
        temp->next = newnode;
    }
}
int addcoff(struct node* head1,struct node* head2){
    int sum=0;
    while(head1!=NULL && head2!=NULL){
        sum+= (head1 -> coff + head2 -> coff);
        head1=head1 -> next;
        head2=  head2  ->  next;
    }
    while(head1!= NULL){
        sum+= head1-> coff;
        head1=head1->  next;
    }
    while(head2!= NULL){
        sum+=head2->coff;
        head2= head2->next;
    }

    return sum;
}
int main(){
    int n,m,coff,power;
    scanf("%d",&n);
    struct node*head[2];
    head[0]=NULL;
    for(int i=0;i<n;i++){
        scanf("%d %d",&coff,&power);
        insert(&head[0],coff,power);
    }
    scanf("%d",&m);
    head[1]=NULL;
    for(int i=0;i<m;i++){
        scanf("%d %d",&coff,&power);
        insert(&head[1],coff,power);
    }
    int result = addcoff(head[0],head[1]);
    printf("%d",result);
    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email:241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arun is learning about data structures and algorithms. He needs your help in solving a specific problem related to a singly linked list.

Your task is to implement a program to delete a node at a given position. If the position is valid, the program should perform the deletion; otherwise, it should display an appropriate message.

Input Format

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

The second line consists of N space-separated elements of the linked list.

The third line consists of an integer x, representing the position to delete.

Position starts from 1.

Output Format

The output prints space-separated integers, representing the updated linked list after deleting the element at the given position.

If the position is not valid, print "Invalid position. Deletion not possible."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
8 2 3 1 7
2

Output: 8 3 1 7

Answer

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

void insert(int);
void display_List();
void deleteNode(int);

struct node {
   int data;
   struct node* next;
} *head = NULL, *tail = NULL;

// You are using GCC
void insert(int data){
   struct node* newnode=(struct node*)malloc(sizeof(struct node));
   newnode->data=data;
   newnode->next=NULL;
   if(head== NULL){
      head=newnode;
   }
   else{
      struct node*temp=head;
```

```
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
    }
}
void deleteNode(int m){
    if(head == NULL||m<=0){
        return;
    }

    struct node* temp=head;
    if(m==1){
        head=temp->next;
        display_List();
        return;
    }
    for(int i=1;temp!=NULL && i<m-1;i++){
        temp=temp->next;
    }
    if(temp==NULL||temp->next == NULL){
        printf("Invalid position. Deletion not possible.");
        return;
    }
    struct node* todelete = temp->next;
    temp->next = temp->next->next;
    display_List();
}
void display_List(){
    struct node* temp=head;
    if(temp)
    while(temp!=NULL){
        printf("%d ",temp->data);
        temp=temp->next;
    }
}


int main() {
    int num_elements, element, pos_to_delete;

    scanf("%d", &num_elements);
    for (int i = 0; i < num_elements; i++) {
        scanf("%d", &element);
        insert(element);
    }

    scanf("%d",  &pos_to_delete);

    deleteNode(pos_to_delete);

    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 3

Attempt : 3
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are working on a text processing tool and need to implement a feature that allows users to insert characters at a specific position.

Implement a program that takes user inputs to create a singly linked list of characters and inserts a new character after a given index in the list.

Input Format

The first line of input consists of an integer N, representing the number of characters in the linked list.

The second line consists of a sequence of N characters, representing the linked list.

The third line consists of an integer index, representing the index(0-based) after which the new character node needs to be inserted.

The fourth line consists of a character value representing the character to be inserted after the given index.

Output Format

If the provided index is out of bounds (larger than the list size):

1. The first line of output prints "Invalid index".
2. The second line prints "Updated list: " followed by the unchanged linked list values.

Otherwise, the output prints "Updated list: " followed by the updated linked list after inserting the new character after the given index.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
a b c d e
2
X
Output: Updated list: a b c X d e

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>
int total=0;
struct node{
    char data;
    struct node *next;
}*head=NULL;
 void create(struct node **head,char data){
    struct node* newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data=data;
    newnode->next=NULL;
    if(*head == NULL){
        newnode->next = *head;
        *head = newnode;
        total++;
    }
    else{
        struct node* temp = *head;
        while(temp->next != NULL){
            temp=temp->next;
        }
       temp->next=newnode;
       total++;
     }
   }
   void display(){
      struct node *temp = head;
      printf("Updated list: ");
      while(temp != NULL){
          printf("%c",temp->data);
          temp=temp->next;
       }
   }
void insert(struct node **head,int place,char data)
{
    struct node* newnode = (struct node*)malloc(sizeof(node));
    newnode->data = data;
    if(place<=total){
        if(*head==NULL && place==0){
            newnode->next=*head;
        }
        else if(*head!= NULL){
            struct node *temp =*head;
            for(int i=0;i<=place-1 && temp->next!= NULL;i++){
                temp=temp->next;
            }
            newnode->next=temp->next;
            temp->next=newnode;
        }
   }
   else{
      printf("Invalid index\n");
   }
```

```
}
int main()
{
    int n;
    char data;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf(" %c",&data);
        create(&head,data);
    }
    int place;
    scanf("%d %c",&place,&data);
    insert(&head,place,data);
    display();
    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

As part of a programming assignment in a data structures course, students are required to create a program to construct a singly linked list by inserting elements at the beginning.

You are an evaluator of the course and guide the students to complete the task.

Input Format

The first line of input consists of an integer N, which is the number of elements.

The second line consists of N space-separated integers.

Output Format

The output prints the singly linked list elements, after inserting them at the beginning.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
78 89 34 51 67

Output: 67 51 34 89 78

Answer

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

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

// You are using GCC
void insertAtFront(struct Node** head,int data){
   struct Node* newnode =(struct Node*)malloc(sizeof(struct Node));
   newnode->data=data;
   newnode->next=*head;
   *head=newnode;
}
void printList(struct Node*head){
   while(head!=NULL){
      printf("%d",head->data);
      head = head->next;
   }
}



int main(){
   struct Node* head = NULL;

   int n;
   scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++) {
        int activity;
        scanf("%d", &activity);
        insertAtFront(&head, activity);
    }

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

    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are tasked with developing a simple GPA management system using a singly linked list. The system allows users to input student GPA values, insertion should happen at the front of the linked list, delete record by position, and display the updated list of student GPAs.

Input Format

The first line of input contains an integer n, representing the number of students.

The next n lines contain a single floating-point value representing the GPA of each student.

The last line contains an integer position, indicating the position at which a student record should be deleted. Position starts from 1.

Output Format

After deleting the data in the given position, display the output in the format "GPA: " followed by the GPA value, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 4
3.8
3.2
3.5
4.1
2
Output: GPA: 4.1
GPA: 3.2
GPA: 3.8
```

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>
struct Node{
   float gpa;
   struct Node* next;
};
void insert(struct Node**head,float gpa){
   struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
   newNode->gpa=gpa;
   newNode->next=*head;
   *head = newNode;
}
void deleteAtPosition(struct Node**head,int position){
   if(*head == NULL)
   return;
   struct Node* temp=*head;
   if(position==1){
      *head = temp->next;
      free(temp);
```

```
        return;
    }
    for(int i=1;temp!=NULL&&i<position -1;i++)
    temp = temp->next;
    if(temp == NULL || temp->next == NULL)
    return;
    struct Node* next=temp->next->next;
    free(temp->next);
    temp->next=next;
}
void display(struct Node* node){
    while(node!=NULL){
        printf("GPA: %.1f\n",node->gpa);
        node =node->next;
    }
}
int main(){
    int n,position;
    struct Node*head=NULL;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        float gpa;
        scanf("%f",&gpa);
        insert(&head,gpa);
    }
    scanf("%d",&position);
    deleteAtPosition(&head,position);
    display(head);
    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John is tasked with creating a program to manage student roll numbers using a singly linked list.

Write a program for John that accepts students' roll numbers, inserts them at the end of the linked list, and displays the numbers.

Input Format

The first line of input consists of an integer N, representing the number of students.

The second line consists of N space-separated integers, representing the roll numbers of students.

Output Format

The output prints the space-separated integers singly linked list, after inserting the roll numbers of students at the end.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
23 85 47 62 31

Output: 23 85 47 62 31

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>
struct node{
   int data;
   struct node*next;
}*head=NULL;
void insert(int data){
   struct node* newnode=(struct node*)malloc(sizeof(node));
   newnode->data=data;
   newnode->next=NULL;
   if(head==NULL){
      head=newnode;
   }
   else{
      struct node *temp=head;
      while(temp->next!=NULL){
         temp=temp->next;
      }
      temp->next=newnode;
   }
}
void display(){
   struct node *temp=head;
   while(temp!=NULL){
      printf("%d ",temp->data);
      temp=temp->next;
   }
}
int main(){
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        int roll;
        scanf("%d",&roll);
        insert(roll);
    }
    display();
    return 0;
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: Jaiyanth T
Email: 241901037@rajalakshmi.edu.in
Roll no:
Phone: 7550097397
Branch: REC
Department: I CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 7

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Dev is tasked with creating a program that efficiently finds the middle element of a linked list. The program should take user input to populate the linked list by inserting each element into the front of the list and then determining the middle element.

Assist Dev, as he needs to ensure that the middle element is accurately identified from the constructed singly linked list:

If it's an odd-length linked list, return the middle element.If it's an even-length linked list, return the second middle element of the two elements.

Input Format

The first line of input consists of an integer n, representing the number of elements in the linked list.

The second line consists of n space-separated integers, representing the elements of the list.

Output Format

The first line of output displays the linked list after inserting elements at the front.

The second line displays "Middle Element: " followed by the middle element of the linked list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50

Output: 50 40 30 20 10
Middle Element: 30

Answer

```
#include  <stdio.h>
#include  <stdlib.h>
struct Node {
   int data;
   struct Node* next;
};


// You are using GCC
struct  Node*  push(struct  Node*  head,int  data)
{
   struct Node* newNode=(struct Node*)malloc(sizeof(struct Node));
   newNode->data=data;
   newNode->next=head;
   return newNode;
}
int printMiddle(struct Node* head)
{
   struct  Node  *slow=head,*fast=head;
```

```
    while(fast!=NULL && fast->next!=NULL)
    {
        slow=slow->next;
        fast=fast->next->next;
    }
    return  slow->data;
}
void  displayList(struct  Node*  head)
{
    struct Node*temp=head;
    while(temp!=NULL)
    {
        printf("%d",temp->data);
        temp=temp->next;
    }
    printf("\n");
}


int main() {
    struct Node* head = NULL;
    int n;

    scanf("%d", &n);
    int value;

    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        head = push(head, value);
    }

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


    int middle_element = printMiddle(head);
    printf("Middle Element: %d\n", middle_element);
    current = head;
    while (current != NULL) {
        struct Node* temp = current;
        current = current->next;
        free(temp);
    }

    return 0;
}
```

Status : Correct *Marks : 10/10*