

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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 0_Arrays and Functions

Attempt : 2
Total Mark : 5
Marks Obtained : 4

Section 1 : Coding

1. 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 arr1[][10], int n) {
```

```
    for( int i = 0; i < n; i++){
```

```
        for( int j = 0; j <= i; j++)
```

```
            { arr1[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

2. 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
```

Status : Wrong

Marks : 0/1

3. 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], int r, int c) {
```

```
    int i,j,sum;
```

```
    for(i=0;i<=r-1;i++)
```

```

    {
        sum =0;
        for(j=0;j<=c-1;j++)
        {
            sum += matrix[i] [j];
        }

        printf("%d",sum);}
    }

```

```

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

4. 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

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

```
int main()
```

```
{
```

```
    int n;
```

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

```
    int salaries[n];
```

```
    int sum = 0, max_salary = 0, min_salary = 1000000;
```

```
    for(int i = 0; i < n; i++)
```

```
    {
```

```
        scanf("%d",&salaries[i]);
```

```
        sum += salaries[i];
```

```
        if(salaries[i] > max_salary){
```

```
            max_salary = salaries[i];
```

```
        }
```

```
        if(salaries[i] < min_salary){
```

```
            min_salary = salaries[i];
```

```
        }
```

```
    }
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        printf("employee %d:%d\n", i+1, salaries[i]);
```

```
    }
```



```
printf("\n");
printf("average salary: %.2f\n", (float)sum/n);
printf("highest salary: %d\n", max_salary);
printf("lowest salary: %d\n", min_salary);
return 0;
}
```

Status : Correct

Marks : 1/1

5. 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

```
#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]);
        }
    }
    int lastsession = matrix[n-1][m-1];
    if (lastsession %2==0)
    {
        printf("%d is even\n",lastsession);
    }
    else
    {
        printf("%d is odd\n",lastsession);
    }
    return 0;
}
```

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 0

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

-

Status : Skipped

Marks : 0/10

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 9

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

```
#include<stdio.h>
#include<stdlib.h>
typedef struct Node{
    char data;
    struct Node* next;
} Node;
Node* createNode(char data){
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}
```

```

Node* createLinkedList(char arr[], int n){
    if(n == 0)return NULL;
    Node* head = createNode(arr[0]);
    Node* temp = head;
    for(int i = 1; i < n; i++){
        temp->next = createNode(arr[i]);
        temp = temp->next;
    }
    return head;
}

```

```

Node* insertAfter(Node* head, int index, char newData, int n){
    /*Node* temp = *head;
    int count = 0;*/
    if (index >= n){
        printf("Invalid index\n");
        printf("Updated list: ");
        Node* temp = head;
        while (temp){
            printf("%c ",temp->data);
            temp = temp->next;
        }
        printf("\n");
        return head;
    }
    Node* temp = head;
    int count = 0;

    while(count < index && temp != NULL){
        temp = temp->next;
        count++;
    }
    Node* newNode = createNode(newData);
    newNode->next = temp->next;
    temp->next = newNode;
    return head;}

void printLinkedList(Node* head){
    printf("Updated list: ");
    Node* temp = head;
    while (temp){
        printf("%c",temp->data);
        if (temp->next) printf(" ");
    }
}

```

```
        temp = temp->next;
    }
    printf("\n");
}
```

```
int main(){
    int n, index;
    char newChar;
    scanf("%d",&n);
    char arr[n];
    for (int i = 0; i < n; i++){
        scanf(" %c",&arr[i]);
    }
    scanf("%d",&index);
    scanf(" %c",&newChar);
    Node* head = createLinkedList(arr,n);
    head=insertAfter(head, index, newChar,n);
    printLinkedList(head);
    return 0;
}
```

Status : Partially correct

Marks : 9/10

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_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;
};
```

```
void insertAtFront(struct Node** head,int value){
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = *head;
    *head = newNode;
}
```

```
void printList(struct Node* head){
    struct Node* temp = head;
    while(temp !=NULL){
        printf("%d",temp->data);
        temp = temp->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: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_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

```
#include<stdio.h>
#include<stdlib.h>
typedef struct Node{
    float gpa;
    struct Node* next;
} Node;
/*Node*newnode(float value){
    Node*newgpa = (Node*)malloc (size of (Node));
    newgpa->value = value
    newgpa->next = NULL;
    return newgpa;
}*/
Node* insertNode(Node* head, float gpa){
    Node* new_Node = (Node*)malloc(sizeof(Node));
    new_Node->gpa = gpa;
    new_Node->next = head;
    return new_Node;
}
Node* deleteNode(Node* head, int position){
    if(head == NULL) return NULL;
```

```
Node* temp = head;
if (position == 1){
    head = head->next;
    free(temp);
    return head;
}
Node* prev = NULL;
for(int i = 1; temp != NULL && i < position; i++){
    prev = temp;
    temp = temp->next;
}
if(temp == NULL) return head;

prev->next = temp->next;
free(temp);
return head;
}

void printList(Node* head){
    Node* temp = head;
    while(temp != NULL){
        printf("GPA: %.1f\n",temp->gpa);
        temp = temp->next;
    }
}

int main(){
    int n, position;
    float gpa;
    Node* head = NULL;
    scanf("%d", &n);
    for(int i = 0; i < n; i++){
        scanf("%f", &gpa);
        head = insertNode(head, gpa);
    }
    scanf("%d", &position);
    head = deleteNode(head, position);
    printList(head);
    return 0;
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_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

```
#include<stdio.h>
#include<stdlib.h>
typedef struct student{
    int roll;
    struct student* next;
}Node;
Node* newNode(int rollno){
    Node* data = (Node*)malloc(sizeof(Node));
    data->roll = rollno;
    data->next = NULL;
    return data;
}
void traverse(Node* head){
    while(head != NULL){
        printf("%d ",head->roll);
        head=head->next;
    }
    printf("\n");
}
int main(){
    int n, rollno;
    scanf("%d", &n);
    if(n<=0) return 0;
    scanf ("%d",&rollno);
    Node* head = newNode(rollno);
    Node* temp = head;
    for (int i=1; i<n; i++){
        scanf("%d",&rollno);
```

```
temp->next=newNode(rollno);  
temp=temp->next;  
}  
traverse (head);  
while (head != NULL){  
    Node* toDelete = head;  
    head = head->next;  
    free(toDelete);  
}  
return 0;  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: ranjani prakash
Email: 240801267@rajalakshmi.edu.in
Roll no: 2116240801267
Phone: 6382555840
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_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;
```

```
};
```

```
struct Node* push(struct Node* head, int value){
```

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

```
    newNode->next = head;
```

```
    newNode->data = value;
```

```
    return newNode;
```

```
}
```

```
int printMiddle(struct Node* head){
```

```
    int len=0;
```

```
    struct Node* temp = head;
```

```
    while(temp !=NULL){
```

```
len++;  
temp = temp->next;  
}  
int pos = len/2;  
for(int i = 0; i<pos;i++){  
    head = head->next;  
}  
return head->data;  
}
```

```
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