

# Rajalakshmi Engineering College

Name: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## 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**

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int coeff;
    int expo;
    struct node*next;
};
typedef struct node node;
node* newnode(int coeff,int expo )
{
    node*newnode=(node*)malloc(sizeof(node));
    newnode->coeff=coeff;
    newnode->expo=expo;
    newnode->next=NULL;
    return newnode;
}
void insert(node**head,int coeff,int expo)
```

```

{
    node*pos=*head;
    if(*head==NULL)
    {
        *head=newnode(coeff,expo);
        return;
    }
    else
    {
        while(pos->next!=NULL)
        {
            pos=pos->next;
        }
        pos->next=newnode(coeff,expo);
    }
}

```

```

int main()
{
    int n;
    scanf("%d",&n);
    int coeff,expo;
    node*poly1;
    node*poly2;
    for(int i=0;i<n;i++)
    {
        scanf("%d%d",&coeff,&expo);
        insert(&poly1,coeff,expo);
    }
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d%d",&coeff,&expo);
        insert(&poly2,coeff,expo);
    }
    int sum=0;
    while(poly1!=NULL)
    {
        sum+=poly1->coeff;
        poly1=poly1->next;
    }
    while(poly2!=NULL)
    {

```

```
    sum+=poly2->coeff;  
    poly2=poly2->next;  
}  
printf("%d",sum);  
  
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

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 : 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
```

```
struct node *pre=NULL;
```

```
void insert(int a)
```

```
{
```

```
    struct node*newnode=(struct node*)malloc(sizeof(struct node));
```

```
    newnode->data=a;
```

```
    newnode->next=NULL;
```

```
    if(head==NULL)
```

```
    {
```

```

        head=newnode;
    }
    else
    {
        tail->next=newnode;}
    tail=newnode;
}
void display_List()
{
    struct node *ptr=head;
    while(ptr!=NULL)
    {
        printf("%d ",ptr->data);
        ptr=ptr->next;
    }
}
void deleteNode(int p)
{
    struct node*ptr=head;
    if(p==1&&head!=NULL)
    {
        head=head->next;
        display_List();
    }

    else
    {
        for(int i=1;i<p&&ptr!=NULL;i++)
        {
            pre=ptr;
            ptr=ptr->next;
        }
        if(ptr==NULL)
        {
            printf("Invalid position. Deletion not possible.");
        }
        else
        {
            pre->next=ptr->next;
            display_List();
        }
    }
}

```

```
}  
}  
}  
  
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: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 3

Attempt : 3  
Total Mark : 10  
Marks Obtained : 0

#### 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***

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

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;  
};
```

```
// You are using GCC
```

```
struct Node *head=NULL,*newnode,*pos;
```

```
void insertAtFront(struct Node **head,int a)
```

```
{  
    newnode=(struct Node*)malloc(sizeof(struct Node));  
    newnode->data=a;  
    newnode->next=NULL;  
    newnode->next=*head;  
    *head=newnode;  
}
```

```
void printList(struct Node*head)
```

```
{  
    pos=head;  
    while(pos!=NULL)  
    {  
        printf("%d ",pos->data);  
        pos=pos->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: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

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 : 0

#### 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**

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 6

Attempt : 3  
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 roll;
    struct node *next;
};
typedef struct node 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;
    }
}
int main()
{
    int n,rollno;
    scanf("%d",&n);
```

```
scanf("%d",&rollno);
Node* head=newnode(rollno);
Node* temp=head;
while(--n)
{
    scanf("%d",&rollno);
    temp->next=newnode(rollno);
    temp=temp->next;
}
traverse(head);
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Nekhita Sri  
Email: 241801184@rajalakshmi.edu.in  
Roll no: 241801184  
Phone: 8637459907  
Branch: REC  
Department: I AI & DS FC  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 7\_CY

Attempt : 1  
Total Mark : 50  
Marks Obtained : 48

### Section 1 : Coding

#### 1. Problem Statement

Arjun is developing a system to monitor environmental sensors installed in different rooms of a smart building. Each sensor records multiple temperature readings throughout the day. To compare sensor data fairly despite differing scales, Arjun needs to normalize each sensor's readings so that they have a mean of zero and standard deviation of one.

Help him implement this normalization using numpy.

Normalization Formula:

#### *Input Format*

The first line of input consists of two integers: sensors (number of sensors) and

samples (number of readings per sensor).

The next sensors lines each contain samples space-separated floats representing the sensor readings.

### **Output Format**

The first line of output prints: "Normalized Sensor Data:"

The next lines print the normalized readings as a numpy array, where each row corresponds to a sensor's normalized values.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 3 3

1.0 2.0 3.0

4.0 5.0 6.0

7.0 8.0 9.0

Output: Normalized Sensor Data:

```
[[ -1.22474487  0.          1.22474487]
```

```
[-1.22474487  0.          1.22474487]
```

```
[-1.22474487  0.          1.22474487]]
```

### **Answer**

```
# You are using Python
```

```
import numpy as np
```

```
n1,n2=map(int,input().split())
```

```
data=[list(map(float,input().split())) for _ in range(n1)]
```

```
df=np.array(data)
```

```
norm=(df - df.mean(axis=1, keepdims=True))/ df.std(axis=1, keepdims=True)
```

```
print("Normalized Sensor Data:")
```

```
print(norm)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Rekha works as an e-commerce data analyst. She receives transaction data containing purchase dates and needs to extract the month and day from these dates using the pandas package.

Help her implement this task by performing the following steps:

Convert the Purchase Date column to datetime format, treating invalid date entries as NaT (missing).

Create two new columns:

Purchase Month, containing the month (as an integer) extracted from the Purchase Date.

Purchase Day, containing the day (as an integer) extracted from the Purchase Date. Keep the rest of the data as is.

### ***Input Format***

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

The second line contains the CSV header — comma-separated column names.

The next  $n$  lines each contain a transaction record in comma-separated format.

### ***Output Format***

The first line of output is the text:

Transformed E-commerce Transaction Data:

The next lines print the pandas DataFrame with:

The original columns (including Purchase Date, which is now in datetime format or NaT if invalid).

Two additional columns: Purchase Month and Purchase Day.

The output uses the default pandas DataFrame string representation as produced by `print(transformed_df)`.

Refer to the sample output for the formatting specifications.

### Sample Test Case

Input: 3

Customer,Purchase Date

Alice,2023-05-15

Bob,2023-06-20

Charlie,2023-07-01

Output: Transformed E-commerce Transaction Data:

	Customer	Purchase Date	Purchase Month	Purchase Day
0	Alice	2023-05-15	5	15
1	Bob	2023-06-20	6	20
2	Charlie	2023-07-01	7	1

### Answer

```
# You are using Python
import pandas as pd
n=int(input())
column=input().split(",")
data=[input().split(",") for _ in range (n)]
df=pd.DataFrame(data, columns=column)
df["Purchase Date"]=pd.to_datetime(df["Purchase Date"],errors='coerce')
df["Purchase Month"]=df["Purchase Date"].dt.month
df["Purchase Day"]=df["Purchase Date"].dt.day
print("Transformed E-commerce Transaction Data:")
print(df)
```

**Status :** Partially correct

**Marks :** 9/10

### 3. Problem Statement

You are working as a data analyst for a small retail store that wants to track the stock levels of its products. Each product has a unique Name (such as "Toothpaste", "Shampoo", "Soap") and an associated Quantity in stock. Management wants to identify which products have zero stock so they can be restocked.

Write a Python program using the pandas library to help with this task. The program should:

Read the number of products, n. Read n lines, each containing the Name of the product and its Quantity, separated by a space. Convert this data into a pandas DataFrame. Identify and display the Name and Quantity of products with zero stock. If no products have zero stock, display: No products with zero stock.

### ***Input Format***

The first line contains an integer n, the number of products.

The next n lines each contain:

<Product\_ID> <Quantity>

where <Product\_ID> is a single word (e.g., "Shampoo") and <Quantity> is a non-negative integer (e.g., 5).

### ***Output Format***

The first line of output prints:

Products with Zero Stock:

If there are any products with zero stock, the following lines print the pandas DataFrame showing those products with two columns: Product\_ID and Quantity.

The column headers Product\_ID and Quantity are printed in the second line.

Each subsequent line shows the product's name and quantity, aligned under the respective headers, with no index column.

The output formatting (spacing and alignment) follows the default pandas to\_string(index=False) style.

If no products have zero stock, print:

No products with zero stock.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 3

P101 10

P102 0

P103 5

Output: Products with Zero Stock:

Product_ID	Quantity
P102	0

### **Answer**

# You are using Python

```
import pandas as pd
```

```
n=int(input())
```

```
data=[input().split() for _ in range(n)]
```

```
df=pd.DataFrame(data, columns=["Product_ID","Quantity"])
```

```
print("Products with Zero Stock:")
```

```
df["Quantity"]=pd.to_numeric(df["Quantity"],errors='coerce')
```

```
zero_rows=df[df["Quantity"]==0]
```

```
if zero_rows.empty:
```

```
    print("No products with zero stock.")
```

```
else:
```

```
    print(zero_rows.to_string(index=False))
```

**Status :** Partially correct

**Marks :** 9/10

## **4. Problem Statement**

Rekha is a meteorologist analyzing rainfall data collected over 5 years, with



monthly rainfall recorded for each year. She wants to find the total rainfall each year and also identify the month with the maximum rainfall for every year.

Help her to implement the task using the numpy package.

Formula:

Yearly total rainfall = sum of all 12 months' rainfall for each year

Month with max rainfall = index of the maximum rainfall value within the 12 months for each year (0-based index)

### ***Input Format***

The input consists of 5 lines.

Each line contains 12 floating-point values separated by spaces, representing the rainfall data (in mm) for each month of that year.

### ***Output Format***

The first line of output prints: yearly\_totals

The second line of output prints: max\_rainfall\_months

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0  
2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0  
3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0  
4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0  
5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0

Output: [ 78. 90. 102. 114. 126.]  
[11 11 11 11 11]

### ***Answer***

```
# You are using Python
import numpy as np
```

```
data=[]
for _ in range(5):
    row=list(map(float,input().split()))
    data.append(row)
df=np.array(data)
total=np.sum(df,axis=1)
max_df=np.argmax(df,axis=1)
print(total)
print(max_df)
```

**Status :** Correct

**Marks :** 10/10

## 5. Problem Statement

Arjun is monitoring hourly temperature data recorded continuously for multiple days. He needs to calculate the average temperature for each day based on 24 hourly readings.

Help him to implement the task using the numpy package.

Formula:

Reshape the temperature readings into rows where each row has 24 readings (one day).

Average temperature per day = mean of 24 hourly readings in each row.

### **Input Format**

The first line of input consists of an integer value,  $n$ , representing the total number of temperature readings.

The second line of input consists of  $n$  floating-point values separated by spaces, representing hourly temperature readings.

### **Output Format**

The output prints: avg\_per\_day

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 30

30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0  
30.0 30.0 30.0 30.0 30.0 30.0 30.0 30.0

Output: [30.]

**Answer**

```
# You are using Python
import numpy as np
n=int(input())
data=list(map(float,input().split()))
df=np.array(data)
reshaped_array=df.reshape(-1, 24)
print(np.mean(reshaped_array,axis=1))
```

**Status :** Correct

**Marks :** 10/10