Rajalakshmi Engineering College

Name: Ranjani S Ranjani S
Email: 241801224@rajalakshmi.edu.in
Roll no: 241801224
Phone: 8489585254
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_PAH_modified

Attempt : 1
Total Mark : 5
Marks Obtained : 3

Section 1 : Coding

1. 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

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
typedef struct Node {
    int coef,exp;
    struct Node* next;
} Node;
int main(){
    int degeree,x,coef;
    Node*head = NULL,*temp;
    scanf("%d",&degeree);
    while(degeree >= 0){
        scanf("%d",&coef);
        temp=(Node*)malloc(sizeof(Node));
        temp->coef=coef;
        temp->exp=degeree--;
        temp->next=head;
        head=temp;
    }
    scanf("%d",&x);
    int result = 0;
    for(temp = head;temp;temp = temp->next)
         result += temp->coef * pow(x,temp->exp);
         printf("%d\n",result);
         return 0;
}
```

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

2. 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 : Skipped ***Marks : 0/1***

3. 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 : Skipped ***Marks : 0/1***

4. 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

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

Status : Correct **_Marks : 1/1_**

5. 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

```
// You are using GCC
#include<iostream>
#include<list>
using namespace std;

void rearrangeList(list<int>& nums){
   list<int> evens,odds;
   for(int num : nums){
      if(num % 2 ==0){
        evens.push_front(num);
        } else {
            odds.push_back(num);
        }
    }
    evens.insert(evens.end(),odds.begin(),odds.end());
    for(auto it = evens.begin(); it != evens.end(); ++it){
        if(it != evens.begin()) cout << " ";
        cout << *it;
    }
    cout << endl;
    }
int main() {
    int n;
    cin >> n;

    list<int> nums;
    for (int i = 0; i < n; i++){
        int val;
        cin >> val;
        nums.push_back(val);
       }

       rearrangeList(nums);
       return 0;
}
```

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

Rajalakshmi Engineering College

Name: Ranjani S Ranjani S
Email: 241801224@rajalakshmi.edu.in
Roll no: 241801224
Phone: 8489585254
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_MCQ

Attempt : 1
Total Mark : 10
Marks Obtained : 9

Section 1 : MCQ

1. Linked lists are not suitable for the implementation of?

Answer

Binary search

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

2. The following function takes a singly linked list of integers as a parameter and rearranges the elements of the lists.

The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node {
   int value;
   struct node* next;
};

void rearrange (struct node* list) {
   struct node *p,q;
   int temp;
   if (! List || ! list->next) return;
   p=list; q=list->next;
   while(q) {
      temp=p->value; p->value=q->value;
      q->value=temp;p=q->next;
      q=p?p->next:0;
   }
}
```

Answer

1, 2, 3, 4, 5, 6, 7

Status : Wrong ***Marks : 0/1***

3. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function.

What should be added in place of "/*ADD A STATEMENT HERE*/", so that the function correctly reverses a linked list?

```
struct node {
   int data;
   struct node* next;
};
static void reverse(struct node** head_ref) {
   struct node* prev   = NULL;
   struct node* current = *head_ref;
   struct node* next;
   while (current != NULL) {
      next  = current->next;
      current->next = prev;
      prev = current;
```

```
        current = next;
    }
    /*ADD A STATEMENT HERE*/
}
```

Answer

*head_ref = prev;

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

4. Given the linked list: 5 -> 10 -> 15 -> 20 -> 25 -> NULL. What will be the output of traversing the list and printing each node's data?

Answer

5 10 15 20 25

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

5. Consider an implementation of an unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operations can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the front node of the linked list

iv) Deletion of the last node of the linked list

Answer

I and III

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

6. Given a pointer to a node X in a singly linked list. If only one point is given and a pointer to the head node is not given, can we delete node X from the given linked list?

Answer

Possible if X is not last node.

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

7. Which of the following statements is used to create a new node in a singly linked list?

```
struct node {
   int data;
   struct node * next;
}
typedef struct node NODE;
NODE *ptr;
```

Answer

ptr = (NODE*)malloc(sizeof(NODE));

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

8. Consider the singly linked list: 13 -> 4 -> 16 -> 9 -> 22 -> 45 -> 5 -> 16 -> 6, and an integer K = 10, you need to delete all nodes from the list that are less than the given integer K.

What will be the final linked list after the deletion?

Answer

13 -&gt; 16 -&gt; 22 -&gt; 45 -&gt; 16

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

9. In a singly linked list, what is the role of the "tail" node?

Answer

It stores the last element of the list

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

10. Consider the singly linked list: 15 -> 16 -> 6 -> 7 -> 17. You need to delete all nodes from the list which are prime.

What will be the final linked list after the deletion?

Answer

15 -&gt; 16 -&gt; 6

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

Rajalakshmi Engineering College

Name: Ranjani S Ranjani S
Email: 241801224@rajalakshmi.edu.in
Roll no: 241801224
Phone: 8489585254
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_week 1_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 28

Section 1 : Coding

1. Problem Statement

Hasini is studying polynomials in her class. Her teacher has introduced a new concept of two polynomials using linked lists.

The teacher provides Hasini with a program that takes two polynomials as input, represented as linked lists, and then displays them together. The polynomials are simplified and should be displayed in the format ax^b, where a is the coefficient and b is the exponent.

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 first line of output prints the first polynomial.

The second line of output prints the second polynomial.

The polynomials should be displayed in the format ax^b, where a is the coefficient and b is the exponent.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1 2
2 1
3 0
3
2 2
1 1
4 0
Output: 1x^2 + 2x + 3
2x^2 + 1x + 4

Answer

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

typedef struct Node {
   int coeff;
   int exp;
   struct Node *next;
} Node;
```

```
Node *createPolynomial(int numTerms){
  Node *head = NULL;
  Node *tail = NULL;
  for (int i=0;i<numTerms;i++){
    int coeff,exp;
    scanf("%d %d",&coeff,&exp);
    Node *newNode = (Node *)malloc(sizeof(Node));
    newNode->coeff = coeff;
    newNode->exp = exp;
    newNode->next = NULL;

    if (head == NULL){
      head = newNode;
      tail = newNode;
    } else{
      tail->next = newNode;
      tail = newNode;
    }
  }
  return head;
}

void displayPolynomial(Node *head){
  Node *current = head;
  int firstTerm = 1;
  while (current != NULL){
    if(current->coeff != 0){
      if(!firstTerm){
        if(current->coeff > 0){
         printf(" + ");
        }else{
          printf(" - ");
          current->coeff = abs(current->coeff);
        }
      }
  if(current->exp == 0){
    printf("%d", current->coeff);
  }else if(current->exp == 1){
    printf("%dx",current->coeff);
  }else{
    printf("%dx^%d",current->coeff,current->exp);
  }
  firstTerm = 0;
  }
  current = current->next;
}
printf("\n");
}

int main(){
  int n,m;
  Node *poly1,*poly2;

  scanf("%d",&n);
  poly1 = createPolynomial(n);

  scanf("%d",&m);
  poly2 = createPolynomial(m);

  displayPolynomial(poly1);
  displayPolynomial(poly2);



  Node *current = poly1;
  while(current != NULL){
    Node *temp = current;
    current = current->next;
    free(temp);
  }
  current = poly2;
  while(current != NULL){
    Node *temp = current;
    current = current->next;
    free(temp);
  }
  return 0;
}
```

Status : Partially correct ***Marks : 8/10***

2. Problem Statement

Keerthi is a tech enthusiast and is fascinated by polynomial expressions. She loves to perform various operations on polynomials.

Today, she is working on a program to multiply two polynomials and delete a specific term from the result.

Keerthi needs your help to implement this program. She wants to take the coefficients and exponents of the terms of the two polynomials as input, perform the multiplication, and then allow the user to specify an exponent for deletion from the resulting polynomial, and display the result.

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 representing the coefficient and the exponent of the term in the first polynomial.

The next line 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 representing the coefficient and the exponent of the term in the second polynomial.

The last line consists of an integer, representing the exponent of the term that Keerthi wants to delete from the multiplied polynomial.

Output Format

The first line of output displays the resulting polynomial after multiplication.

The second line displays the resulting polynomial after deleting the specified term.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
2 2
3 1
4 0
2
1 2
2 1
2

Output: Result of the multiplication: 2x^4 + 7x^3 + 10x^2 + 8x
Result after deleting the term: 2x^4 + 7x^3 + 8x

Answer

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

typedef struct {
   int coeff;
   int exp;
}Term;
void addTerm(Term poly[],int *size,int coeff,int exp){
   for (int i=0;i<*size;i++){
      if(poly[i].exp == exp){
         poly[i].coeff += coeff;
         return;
      }
   }
   poly[*size].coeff = coeff;
   poly[*size].exp = exp;
   (*size)++;
}
void displayPolynomial(Term poly[],int size){
   for(int i=0;i<size;i++){
      if(poly[i].coeff == 0)
      continue;
      if(i>0)
      printf(" + ");
      if(poly[i].exp == 1){
      printf("%dx",poly[i].coeff);
      } else if (poly[i].exp == 0){
         printf("%d",poly[i].coeff);
      } else{
      printf("%dx^%d",poly[i].coeff,poly[i].exp);
```

```
    }
    }
    printf("\n");
}
int main(){
    int n,m,delExp;
    Term poly1[5],poly2[5],result[25];
    int resultSize = 0;
    scanf("%d",&n);
    for (int i=0;i<n;i++){
        scanf("%d %d",&poly1[i].coeff,&poly1[i].exp);
    }
    scanf("%d",&m);
    for(int i=0;i<m;i++)   {
        scanf("%d %d",&poly2[i].coeff,&poly2[i].exp);
    }
    scanf("%d",&delExp);

    for(int i=0;i<n;i++){
        for(int j=0;j<m; j++){
            addTerm(result,&resultSize,poly1[i].coeff * poly2[j].coeff,poly1[i].exp +
poly2[j].exp);
        }
    }
    printf("Result of the multiplication: ");
    displayPolynomial(result,resultSize);
    scanf("%d",&delExp);
    Term finalResult[25];
    int finalSize = 0;
    for (int i=0;i<resultSize;i++){
        if(result[i].exp !=delExp)
        {
            finalResult[finalSize++] = result[i];
        }
    }
            printf("Result after deleting the term: ");
            displayPolynomial(finalResult,finalSize);

            return 0;
        }
```

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

3. Problem Statement

Timothy wants to evaluate polynomial expressions for his mathematics homework. He needs a program that allows him to input the coefficients of a polynomial based on its degree and compute the polynomial's value for a given input of x. Implement a function that takes the degree, coefficients, and the value of x, and returns the evaluated result of the polynomial.

Example

Input:

degree of the polynomial = 2

coefficient of x2 = 13

coefficient of x1 = 12

coefficient of x0 = 11

x = 1

Output:

36

Explanation:

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

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

Calculate the value of 11x0: 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 an integer representing the degree of the polynomial.

The second line consists of an integer representing the coefficient of x2.

The third line consists of an integer representing the coefficient of x1.

The fourth line consists of an integer representing the coefficient of x0.

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

Output Format

The output is an 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

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

int main(){
   int degeree;
   scanf("%d",&degeree);
   int coeff[degeree + 1];
   for(int i=degeree; i>=0;i--){
      scanf("%d",&coeff[i]);
   }
   int x;
   scanf("%d",&x);
   int result = 0;


   for(int i= degeree; i>=0; i--){
      result += coeff[i] * pow(x, i);
   }
   printf("%d",result);
```

```
    return 0;
}
```

Status : Correct

Marks : 10/10