

Practical 1

DOMS Page No.
Date / /

Aim : Multiply two matrices.

```
#include <iostream>
using namespace std;

int main()
{
    int a[10][10], b[10][10], mult[10][10], r1, c1, r2, c2, i, j, k;
    cout << "Enter rows and columns for first matrix : ";
    cin >> r1 >> c1;
    cout << "Enter rows and columns for second matrix : ";
    cin >> r2 >> c2;
    while (c1 != r2)
    {
        cout << "Error! Column of first matrix not equal to
row of second. ";
        cout << "Enter rows and columns for first matrix : ";
        cin >> r1 >> c1;

        cout << "Enter rows and columns for second matrix : ";
        cin >> r2 >> c2;
    }

    cout << endl << "Enter elements of matrix 1 : " << endl;
    for (i = 0; i < r1; ++i)
        for (j = 0; j < c1; ++j)
    {
        cout << "Enter element a" << i + 1 << j + 1 << " : ";
        cin >> a[i][j];
    }
}
```

```

cout << "Enter elements of matrix 2:" << endl;
for(i=0 ; i<r2 ; ++i)
    for(j=0 ; j<c2 ; ++j)
{
    cout << "Enter element b" << i+1 << j+1 << " : ";
    cin >> b[i][j];
}
for(i=0 ; i<r1 ; ++i)
    for(j=0 ; j<c2 ; ++j)
{
    mult[i][j]=0;
}
for(i=0 ; i<r1 ; ++i)
    for(j=0 ; j<c2 ; ++j)
        for(k=0 ; k<c1 ; ++k)
{
    mult[i][j] += a[i][k] * b[k][j];
}
cout << "Resultant Matrix : \n";
for(i=0 ; i<r1 ; ++i)
    for(j=0 ; j<c2 ; ++j)
{
    cout << " " << mult[i][j];
    if(j == c2-1)
        cout << endl;
}
return 0;
}

```

Practical 2

DOMS Page No.
Date / /

Aim : Add and Subtract array using switch case

```
#include<iostream>
using namespace std;
class Matrices
{
    int i, j, a[10][10], b[10][10], add[10][10], sub[10][10],
        r, c;
public:
    void RangeM()
    {
        cout << "Enter the range of matrix : ";
        cin >> r >> c;
        cout << "\nEnter the element for 1 Matrix \n";
        for (i=0; i<r; i++)
        {
            for (j=0; j<c; j++)
            {
                cin >> a[i][j];
            }
        }
        cout << "\nEnter the element for 2 Matrix \n";
        for (i=0; i<r; i++)
        {
            for (j=0; j<c; j++)
            {
                cin >> b[i][j];
            }
        }
    }
}
```

```

void display()
{
    cout << " First Matrix \n";
    for (i=0 ; i<r ; i++)
    {
        for (j=0 ; j<c ; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << "\n";
    }
    cout << " Second Matrix \n";
    for (i=0 ; i<r ; i++)
    {
        for (j=0 ; j<c ; j++)
        {
            cout << b[i][j] << " ";
        }
        cout << "\n";
    }
}

void Add-Sub-Mat()
{
    for (i=0 ; i<r ; i++)
    {
        for (j=0 ; j<c ; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
            d[i][j] = a[i][j] - b[i][j];
        }
    }
}

```

```
Void display - Add()
{
    cout << "\n Addition ---- \n";
    for (i=0; i<8; i++)
    {
        for (j=0; j<c; j++)
        {
            cout << cc[i][j] << " ";
        }
        cout << "\n";
    }
}

Void display - Sub()
{
    cout << "\n Subtraction .... \n";
    for (i=0; i<8; i++)
    {
        for (j=0; j<c; j++)
        {
            cout << d[i][j] << " ";
        }
        cout << "\n";
    }
}

int main()
{
    system("clear");
    Matrices m;
    m. displayM();
    int choose;
```

```
while(1)
{
```

```
    cout << "Choose the Action You Want to Perform  
(1-1. Addition \n 2. Subtraction \n 3. Quit \n"
```

```
    choice : ";
```

```
    cin >> choose;
```

```
    switch (choose)
{
```

```
    Case 1:
```

```
        m. display - Add();  
        break;
```

```
    Case 2:
```

```
        m. display - Sub();  
        break;
```

```
    Case 3:
```

```
        exit (0);
```

```
    default:
```

```
        cout << " Sorry You have entered wrong choice...  
                \n";
```

```
}
```

```
return 0;
```

```
}
```

Enter the element of 1D Array
2
3
4
5
6

Enter the element to search 4
found at index 2

OR

Enter the element of 1D Array
2
5
4
1
0

Enter the element to search 3
not found

Practical 3

18/8/23

- WAP++ prog to perform Linear Search Operation on a 1D Array.

```
#include <iostream>
using namespace std;
int main()
{
    int a[5], n, flag=0;
    cout << "Enter the element of 1D array ";
    for(int i=0; i<5; i++)
    {
        cin >> a[i];
    }
    cout << "Enter the element to search \n";
    cin >> n;
    for(int i=0; i<5; i++)
    {
        if(a[i]==n)
        {
            cout << "found at index " << i;
            flag=1;
            break;
        }
    }
    if(flag==0)
    {
        cout << "not found";
    }
    return 0;
}
```

FOR EDUCATIONAL USE

Enter no. of element in the Array : 5

Enter elements
11
22
33
55
01

the elements are:
11 22 33 55 01

Enter no. to search 45

element not found.

Practical 4

25/8/23

* Write to perform binary Search operation on 1D Array.

Using namespace std;
int main()
{

```
int a[20], start, size, end, mid, num, i;
cout << "Enter no. of element in the Array : ";
cin >> size;
cout << "Enter elements \n";
for(i=0; i<size; i++)
{
    cin >> a[i];
}
```

```
cout << "the elements are : \n";
for(i=0; i<size; i++)
{
    cout << a[i] << " ";
}
```

```
cout << "\n Enter no. to search\n";
cin >> num;
```

```
start = 0;
end = size - 1;
mid = (start + end) / 2;
while (start <= end)
{
    if (a[mid] == num)
```

```
    {
        cout << "element found";
        break;
    }
}
```

FOR EDUCATIONAL USE

```
else if (a[mid] < num)
{
    start = mid + 1;
}
else
{
    end = mid - 1;
}
mid = (start + end) / 2;
if (start > end)
{
    cout << "element not found";
}
return 0;
}
```

Enter no. of element in the array : 4

Enter elements

1212

3411

300

2604

the elements are :

1212 3411 300 2604

Sorting

the element after Sorting are :

300 1212 2604 3411

Practical 3

- WAPC++ Code to Sort the element of 1D Array using Bubble Sort algorithm

```
→ int main()
{
    int a[50], start, end, size, mid, num, i, temp;
    cout << "Enter no. of element in the array: ";
    cin >> size;
    cout << "Enter elements \n";
    for (i=0 ; i<size ; i++)
    {
        cin >> a[i];
    }
    cout << "the elements are: \n";
    for (i=0 ; i<size ; i++)
    {
        cout << a[i] << " ";
    }
    cout << "\n Sorting \n";
    for (i=0 ; i<size ; i++)
    {
        for (int j=0 ; j<size-i ; j++)
        {
            if (a[j]>a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}
```

P.

```
cout << "the element after sorting are : "
for (i=1 ; i<=size ; i++)
{
    cout << a[i] << " "
```

the 1st element is:

7

8

Practical 6A

o/g/23
• Write Prgm to insert an element at the start of
Singly linked list

```
#include <iostream>
using namespace std;
// creation of structure of node
// user defined data type node
class Node {
public:
    int info;
    Node *next;
};

Void insertATStart (Node ** begin ,int data)
{
    Node *newNode = newNode(); //empty Node.
    newNode->info = data;
    newNode->next = *begin;
    *begin = newNode;
}

Void displaySLL (Node * ptr) {
    while (ptr != NULL) {
        cout << " the 1st element is : " << ptr->info;
        ptr = ptr->next;
        cout << endl;
    }
}

int main() {
    Node * begin = NULL ;
    insertATStart (& begin , 7);
    insertATStart (& begin , 8);
    displaySLL (begin);
    return 0;
}
```

FOR EDUCATIONAL USE

the 1st element is

7

8

9

Practical 6B

o WActual code to insert an element at the end of SLL
01/09/23
→ #include <iostream>
using namespace std;
class Node {
public :
 int info;
 Node * next;
};
void insertAtEnd (Node ** begin, int data) {
 Node * newNode = new Node();
 newNode->info = data;
 newNode->next = NULL;
 if (*begin == NULL) {
 *begin = newNode;
 }
 else {
 Node * ptr = *begin;
 while (ptr->next != NULL) {
 ptr = ptr->next;
 }
 ptr->next = newNode;
 }
}
void displaySLL (Node * ptr) {
 while (ptr != NULL) {
 cout << "the 1st element is " << ptr->info
 << endl;
 ptr = ptr->next;
 }
}

Aundaram®

FOR EDUCATIONAL USE

```
int main() {  
    Node *begin = NULL;  
    insertATEnd (&begin, 7);  
    insertATEnd (&begin, 8);  
    insertATEnd (&begin, 9);  
    displaySLC (begin);  
    return 0;  
}
```

Linked List is :

10 9 8 7 6 5 4

Value deletion is : 10

Value deletion is : 9

Linked List : 8 7 6 5 4

Practical 6C

05/01/23

- WAP to Code to delete an element at the start of SLL.

```
#include <iostream>
using namespace std;
class Node {
public:
    int info;
    Node *next;
};
```

```
void insert(Node **begin, int data)
```

```
{ Node *newNode = new Node();
    newNode->info = data;
    newNode->next = *begin;
    *begin = newNode;
}
```

```
void display(Node *ptr)
{
```

```
    cout << "In Linked List is :\n";
    while (ptr != NULL)
        cout << ptr->info << " ";
    ptr = ptr->next;
}
```

```
void deleteAtStart(Node **begin)
```

```
{ Node *old = *begin;
```

```
// L.L is empty
```

```
if (*begin == NULL)
```

```
{ cout << "In empty L.L , deletion not possible \n";
```

```
return;
```

FOR EDUCATIONAL USE

```
// L.L contain Sonly 1 Node  
if (Old->next == NULL)  
{  
    cout << "In value deletion is :" << Old->info;  
    * begin = NULL;  
    return;  
}  
// If L.L Contains more than 1 Node  
* begin = Old-> next;  
cout << "In value deletion :s " << Old-> info;  
// free (*Old);  
  
int main() {  
    Node * begin = NULL;  
    insert (&begin , 4);  
    insert (&begin , 5);  
    ::  
    display (begin);  
    deleteAtStart (&begin);  
    deleteAtStart (&begin);  
    display (begin);  
    return 0;  
}
```

5/9/23

10 min

Linked list is : 10 9 8 7 6 5 4

Value deleted is 4
Value deleted is 5
Value deleted is 6

Linked list is : 10 9 8 7

Practical 6D

5/9/23

- WAP to code to delete an element at the end of SLL

```
#include <iostream>
using namespace std;
class Node {
public:
    int info,
        Node *next;
};

void insert(Node **begin, int data) {
    newNode->info = data;
    newNode->next = *begin;
    *begin = newNode;
}

void display(Node *ptr) {
    cout << "\n Linked list is : ";
    while (ptr != NULL) {
        cout << ptr->info << " ";
        ptr = ptr->next;
    }
}

void deleteEnd(Node **begin) {
    Node *old = *begin;
    Node *prev;
    // LL is empty
    if (*begin == NULL) {
        cout << "\n Empty L.L, deletion not possible \n";
        return;
    }
}
```

FOR EDUCATIONAL USE

GeForce 1650
NVIDIA

```
// L.L contains only 1 node  
if (old->next == NULL) {  
    cout << "In Value deleted is " << old->info;  
    *begin = NULL;  
    return;  
}  
while (old->next != NULL) {  
    prev = old;  
    old = old->next;  
}  
prev->next = NULL;  
cout << "In Value deleted is " << old->info;  
int main() {  
    Node *begin = NULL;  
    insert(&begin, 4);  
    insert(&begin, 5);  
    insert(&begin, 6);  
    insert(&begin, 7);  
    insert(&begin, 8);  
    insert(&begin, 9);  
    display(begin);  
    deleteEnd(&begin);  
    deleteEnd(&begin);  
    deleteEnd(&begin);  
    display(begin);  
    return 0;  
}
```

1 Inserted
2 Inserted
3 Inserted
4 Inserted
5 Inserted

Linked List :

5 4 3 2 1

PRACTICAL 7A

DOMS Page No.
Date / / 14/9/23

- WAP C++ code to perform Insertion at the start of Circular Linked List.

```
#include <iostream>
using namespace std;
class Node {
public:
    int info;
    Node *next;
};
void insertStart (Node **begin, int data)
{
    Node *newNode = new Node(); // empty
    newNode->info = data; // insert Data
    // if first node is being inserted
    if (*begin == NULL)
    {
        *begin = newNode;
        (*begin)->next = *begin;
        cout << newNode->info << " Inserted" << endl;
        return;
    }
    // if had more than 1 node do below
    Node *ptr = *begin;
    // traverse till Last node in circular linked list
    while (ptr->next != *begin)
    {
        ptr = ptr->next;
    }
}
```

```
ptr -> next = newNode;
newNode -> next = * begin;
cout << newNode -> info << " Inserted" << endl;
* begin = newNode;
}
void display(Node * begin) {
    cout << "In Linked List : " << endl;
    // if circular linked list is empty currently
    if (begin == NULL)
        return;
    Node * temp = begin;
    // Since we need to take care of circular nature
    // of Linked List
    do {
        cout << temp -> info << " ";
        temp = temp -> next;
    }
    while (temp != begin);
}
cout <<
```

```
int main()
{
    // first node will be NULL at creation
    Node *begin = NULL;
    insertStart(&begin, 1);
    insertStart(&begin, 2);
    insertStart(&begin, 3);
    insertStart(&begin, 4);
    insertStart(&begin, 5);
    display(begin);
    return 0;
}
```

-* -

1 Inserted
2 Inserted
3 Inserted
4 Inserted
5 Inserted

Linked List :
1 2 3 4 5

Practical FB

DOMS Page No.
Date / /

Aim : Insertion at the end of circular linked list.

```
#include <iostream>
using namespace std;
class Node
{
public:
    int info;
    Node *next;
};

void InsertEnd(Node **begin, int data)
{
    Node *newNode = new Node(); //empty Node
    newNode->info = data;
    //if first node is being inserted
    if (*begin == NULL)
    {
        *begin = newNode;
        (*begin)->next = *begin;
        cout << newNode->info << " Inserted" << endl;
        return;
    }
    //if had more than 1 node
    Node *ptr = *begin;
    //traverse till last node in circular LL
    while (ptr->next != *begin)
```

DOMS | Page No.
Date / /

```
{  
    ptr = ptr->next;  
}  
ptr->next = newNode;  
newNode->next = *begin;  
cout << newNode->info << " Inserted \n";  
}  
Void display (Node * begin)  
{  
    cout << "In Linked List : \n";  
    //if circular L.L is empty currently  
    if (begin == NULL)  
        return;  
    Node * ptr = begin;  
    //Since we need to take care of Circular nature of L.L  
    do  
    {  
        cout << ptr->info << " ";  
        ptr = ptr->next;  
    }  
    while (ptr != begin);  
    cout << endl;  
}  
int main()  
{  
    Node * begin = NULL;  
    InsertEnd (&begin, 1);  
    InsertEnd (&begin, 2);  
    InsertEnd (&begin, 3);  
    InsertEnd (&begin, 4);  
}
```

```
insertEnd(&begin, 5);  
display(begin);  
return 0;
```

3

—

Enter a positive integer : 5
factorial of 5 = 120

Practical 8A

DOMS Page No:
Date / /

Aim: WAP code in C++ to find the factorial of a number using recursion.

```
#include <iostream>
using namespace std;

int factorial(int n);
int main()
{
    int n;
    cout << "Enter a positive integer : ";
    cin >> n;
    cout << "Factorial of " << n << " = " << factorial(n);
    return 0;
}

int factorial(int n)
{
    if (n >= 1)
        return n * factorial(n - 1);
    else
        return 1;
}
```

Input the number of terms for fibonacci series : 7
Fibonacci Series is :

0 1 1 2 3 5 8

Practical 8B

DOMS Page No.
Date / / 18/10/22

Aim : WA code in C++ to display Fibonacci series using recursion ...

```
#include <iostream>
using namespace std;
int fibo(int n);
int main()
{
    int n, i = 0,
        cout << "Input the number of terms for Fibonacci series : ";
    cin >> n;
    cout << "\nFibonacci Series is :\n";
    while (i < n)
    {
        cout << " " << fibo(i);
        i++;
    }
    return 0;
}
int fibo(int n)
{
    if (n == 1 || n == 0)
    {
        return n;
    }
    else
    {
        return fibo(n-1) + fibo(n-2);
    }
}
```