Day 6

10 April 2024 01:03 PM

Sorting :→ Selection Insertion

n 5

Pass-1 a 5 1 4 3 2

Pass-2 a 1 2 4 3 3

Pass-3 a 1 2 3 4 4

Pass-4 a 1 2 3 4 5

```
if (a[i] > a[j])
    swap(a[i], a[j])
```

i	j
0	1-4
1	2-4
2	3-4
3	4-4

i = 0 to < n-1 ++

j = i+1 to < n ++

a 1 2 3 4 5

Sorted Array

$n-1 + n-2 + n-3 \cdots 1 \rightarrow \frac{n * n-1}{2}$

$O(n^2)$

$\frac{n^2 + n}{2}$

a 1 2 3 4 5

i 10

i = 10

i = a[0]

auto

auto i = 10;

auto i = 7.5f;

i 10

i < a.length()

object

```
for(auto i : a)
        cout<< i <<" ";
```

Insertion →

item 6

a	2	3	4	5	6	8
→	0	1	2	3	4	5

int i, j, item

```
for(i=1 ; i<n ; i++)
{
    int item = a[i];
    for(j=i-1 ; j>=0 and a[j]>item ; j--)
    {
        a[j+1] = a[j];
    }
    a[j+1] = item ;
}
```

Best Case

1	2	3	4	5

O(n)

worst case

a	5	4	3	2	1

$O(n^2)$

int a;

int a[5];

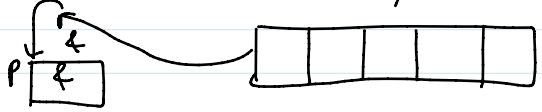
Dynamic Array :-

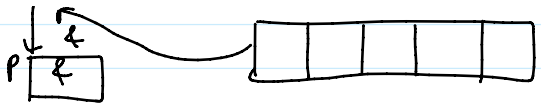
{ new
  delete

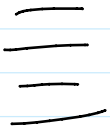
n = 5

int *p;

p = new int[n];





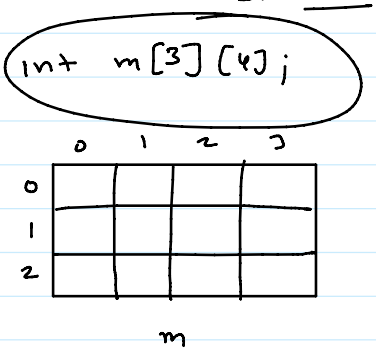


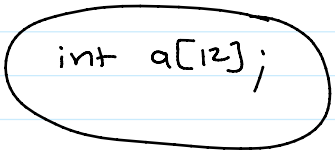
delete [] p;

```
#include<iostream>
using namespace std;
int main()
{
    int *arr;
    int n;
    cout<<"Enter no of elements:";
    cin>>n;
    //dynamic memory
    arr=new int[n];
    //input
    for(int i=0;i<n;i++)
    {
        cout<<"Enter value of "<<i+1<<" element:";
        cin>>arr[i];
    }
    //output
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
    //free dynamic memory
    delete []arr;
    return 0;
}
```

2-D Array :→

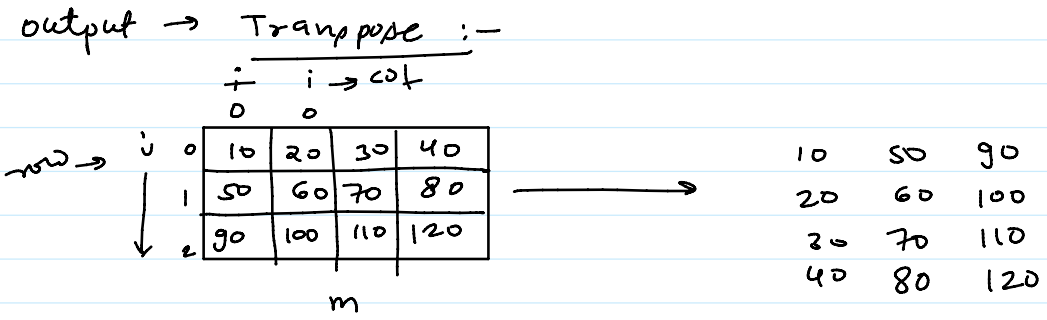
Static






```
#include<iostream>
using namespace std;
int main()
{
    int m[10][10];
    int r,c;
    cout<<"Enter value of row and cols:";
```

```
    cin>>r>>c;
    //input
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<"Enter value of["<<i+1<<"]["<<j+1<<"] element";
            cin>>m[i][j];
        }
    }
    //output
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<m[i][j]<<"\t";
        }
        cout<<endl;
    }
    return 0;
}
```

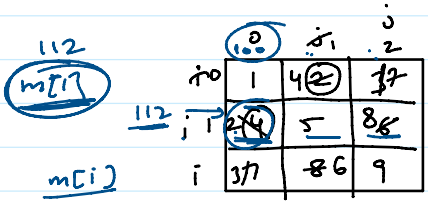


m[j][i]

```
#include<iostream>
using namespace std;
int main()
{
    int m[10][10];
    int r,c;
    cout<<"Enter value of row and cols:";
    cin>>r>>c;
    //input
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<"Enter value of["<<i+1<<"]["<<j+1<<"] element";
            cin>>m[i][j];
        }
    }
    //output
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<m[i][j]<<"\t";
        }
        cout<<endl;
    }
    //Transpose output
    cout<<"Transpose Output\n";
```

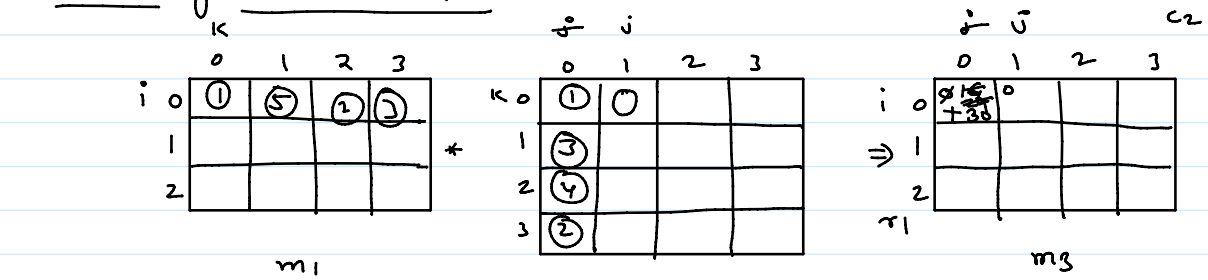
```
        for(int i=0;i<c;i++)
        {
            for(int j=0;j<r;j++)
            {
                cout<<m[j][i]<<"\t";
            }
            cout<<endl;
        }
        return 0;
    }
```

int a[3][4]




```
for(i=0; i<r; i++)
{
    for(j=i+1; j<c; j++)
    {
        swap(a[i][j], m[j][i])
    }
}
```

Multiply 2 Matrices :-




```
m3[i][j] += m1[i][k] * m2[k][j];
```

```
for(i=0; i<r1; i++)
{
    for(j=0; j<c2; j++)
    {
        m3[i][j]=0;
        for(k=0; k<c1; k++)
        {
            m3[i][j] += m1[i][k] * m2[k][j];
        }
    }
}
```

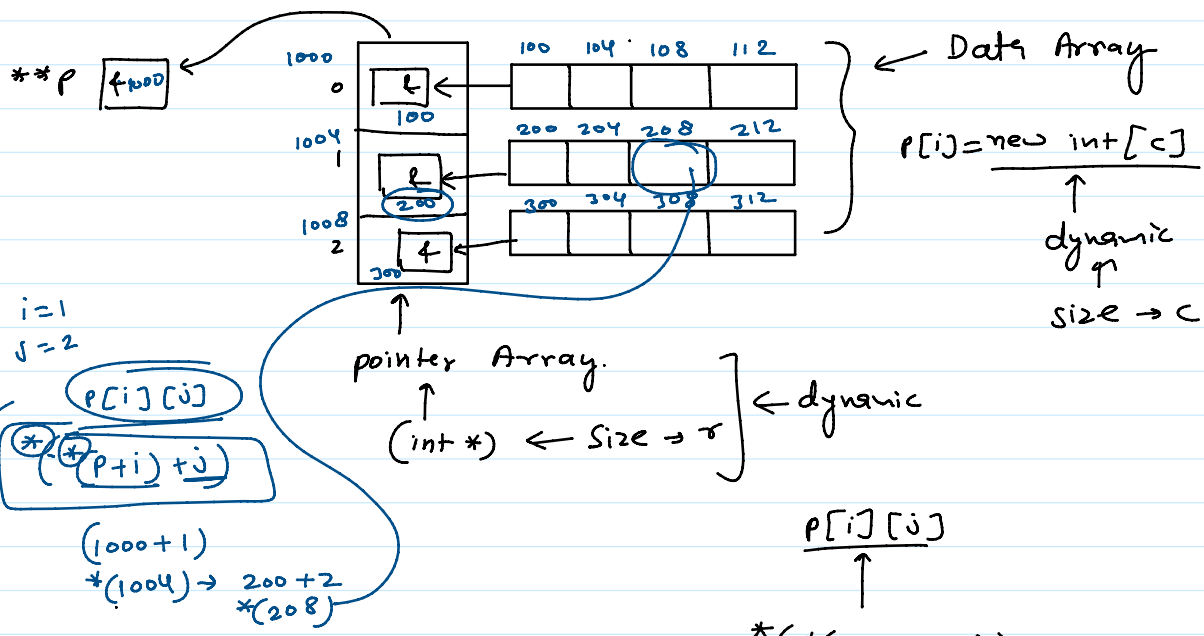
```
if(c1 != r2)
{
    return 1;
}
```

Dynamic matrix :→

r = 3
c = 4

Dynamic matrix :→

r = 3
c = 4



i=1
j=2

p[i][j]

((p+i)+j)

(1000+1)
*(1004) → 200+2
*(208)

p[i][j]

((p+i)+j)

```
#include<iostream>
using namespace std;
int main()
{
    int **p;
    int r,c;
    cout<<"Enter no of row ans cols:";
    cin>>r>>c;
    //Dynamic memory
    p=new int*[r];  //pointer array
    for(int i=0;i<r;i++)
        p[i]=new int[c];    //data array
    //input
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<"Enter ["<<i+1<<"]["<<j+1<<"]:";
            cin>>p[i][j];
        }
    }
    //output
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<p[i][j]<<'\t';
        }
        cout<<endl;
    }
    //free dynamic memory
    for(int i=0;i<r;i++)
        delete []p[i];      //data array
    delete []p;     //pointer array
    return 0;
}
```

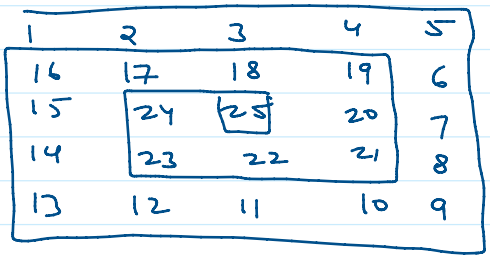
vector <vector <int>>

n 5

H.W

1	2	3	4	5
16	17	18	19	6

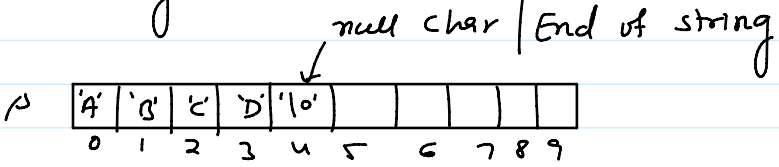
H.W



String :-

```
char A[10] = "ABCD";
```

A → string




```
#include<iostream>
using namespace std;
int main()
{
    char s[20];
    cout<<"Enter your name:";
    //cin>>s;
    cin.getline(s,20);
    cout<<"Your name is "<<s;
    return 0;
}
```

<string.h> → strlen()

```
char A[10] = "ABCD";
cout<< strlen(A); → 4
```

strcpy() :→

```
char s1[10] = "ABCD";
char s2[10];
s2 = s1;   X
strcpy(s2, s1);
```

```
char s1[10];
.
.
.
s1 = "ABCD";   X
strcpy(s1, "ABCD");
```

strcat :→

```
char s1[10] = "ABCD";
char s2[10] = "XYZ";
strcat(s1, s2);
```

(Append)

cout<<s1; → ABCDXYZ.

strcmp() :→

```
char s1[10] = "ABCD";
char s2[10] = "AYZ";
```

cout<<strcmp(s1, s2); → -1

$s_1 > s_2 \rightarrow 1$

$s_1 == s_2 \rightarrow 0$

$s_1 < s_2 \rightarrow -1$

strrev() :-

```
char A[10] = "ABCD";
strrev(A);
cout<<s;
```

→ DCBA

string class :→

#include <string>

```
string s = "ABCD";
```

string → Data type

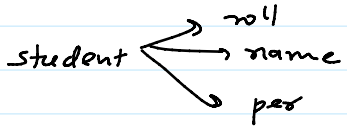
s → object

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string s;
    cout<<"Enter your name:";
    //cin>>s;
    getline(cin,s);
    cout<<"Your name is "<<s;
```

```
    return 0;
}
```

structure :→ user defined data type

struct keyword →



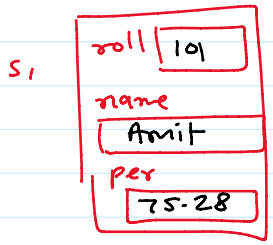
Syntax →

```
struct name
{
    member1;
    member2;
    .
    .
};
```

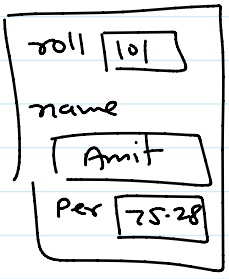
```
struct student
{
    int roll;
    char name[20];
    float per;
};
```

```
student s1;
```

Data type — object



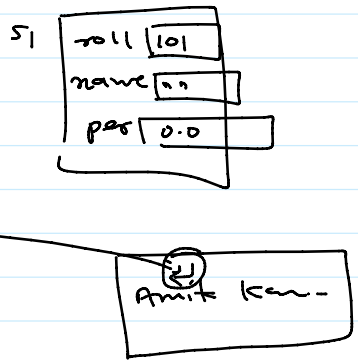

```
s1.roll = 101;
s1.name = "Amit";
strcpy(s1.name, "Amit");
s1.per = 75.28;
```




```
student s2;
.
.
s2 = s1;
```

```cpp
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string s;
    cout<<"Enter your name:";
    //cin>>s;
    getline(cin,s);
    cout<<"Your name is "<<s;
    return 0;
}
```

```cpp
#include<iostream>
#include<string>
using namespace std;
struct student
{
    int roll;
    char name[20];
    float per;
};
int main()
{
    student s1;
    cout<<"Enter roll name and per of a student:";
    cin>>s1.roll;
    cin.getline(s1.name,20);
    cin>>s1.per;
    cout<<s1.roll<<"\t"<<s1.name<<'\t'<<s1.per<<endl;
    return 0;
}
```




```cpp
#include<iostream>
#include<string>
using namespace std;
struct student
{
    int roll;
    char name[20];
    float per;
};
int main()
{
    student s1;
    cout<<"Enter roll name and per of a student:";
    cin>>s1.roll;
    cin.ignore();
    cin.getline(s1.name,20);
    cin>>s1.per;
    cout<<s1.roll<<"\t"<<s1.name<<'\t'<<s1.per<<endl;
    return 0;
}
```