HW-4

Hammad Zafar (19-ee-328)

```
#include <iostream>
#include <new>
using namespace std;
struct matrix
{
   int r,c,*p,k;
};
void input_data(int n,matrix mat[],string &m);
void m_a(int n,matrix mat[]);
void m_s(int n,matrix mat[]);
void print(int n, matrix mat[],string m);
int main()
{
         string m;
         int n=4;
         matrix mat[4];
   input_data(n,mat,m);
   m_a(n,mat);
   m_s(n,mat);
   print(n,mat,m);
   for(int i=0;i<n;i++)
   {
         delete [] mat[i].p;
}
}
```

```
void input_data(int n,matrix mat[],string &m)
{

        cout<<"Dear user,please enter your name: \n";
        cin>>m;
        cout<<"Dear "<<m<<",enter the dimensions  of X matrix: \n";
        cin>>mat[0].r;
        cin>>mat[0].c;
        cout<<"Dear "<<m<<",enter the contents of X matrix  \n";
        mat[0].p= new int[mat[0].r * mat[0].c];
        for(int i=0;i<mat[0].r;i++)
        {
                for(int j=0;j<mat[0].c;j++)
                {
                        cin>>mat[0].k;
                        *(mat[0].p+i*mat[0].c+j)=mat[0].k;
                }
        }
        cout<<"Dear "<<m<<" ,enter the dimensions of Y matrix\n";
        cin>>mat[1].r;
        cin>>mat[1].c;
        cout<<"Dear "<<m<<" ,enter the contents of Y matrix \n";
        mat[1].p= new int[mat[1].r * mat[1].c];
        for(int i=0;i<mat[1].r;i++)
        {
                for(int j=0;j<mat[1].c;j++)
                {
                        cin>>mat[1].k;
                        *(mat[1].p+i*mat[1].c+j)=mat[1].k;
                }
        }
}
void m_a(int n,matrix mat[])
 {
        int i,j,a,b,s;
   try
   {
          if(mat[0].r==mat[1].r and mat[0].c==mat[1].c)
        {
                mat[2].r=mat[0].r;
                mat[2].c=mat[0].c;
        mat[2].p= new int[mat[2].r * mat[2].c];
        for(int i=0;i<mat[0].r;i++)
          {
                for(int j=0;j<mat[0].c;j++)
                {
                        a=*(mat[0].p + i*mat[0].c +j);
                        b=*(mat[1].p + i*mat[1].c +j);
                        s=a+b;
                        *(mat[2].p+ i*mat[2].c+j)=s;
                }
           }
 }
   else throw(mat[0].c);
}
   catch (...)
   {
           cout<<"dimensions are not same,so addition  not possible\n";
   mat[2].r=0;
   mat[2].r=0;
           return;}
}

void m_s(int n,matrix mat[])
{
           int a,b,s;
           try
           {
   if((mat[0].r==mat[1].r) && (mat[0].c==mat[1].c))
           {
                       mat[3].r=mat[0].r;
                       mat[3].c=mat[0].c;
               mat[3].p= new int[mat[3].r * mat[3].c];
           for(int i=0;i<mat[0].r;i++)
           {

                       for(int j=0;j<mat[0].c;j++)
                       {
                                   a=*(mat[0].p + i*mat[0].c +j);
                                   b=*(mat[1].p + i*mat[1].c +j);
                                   s=a-b;
                                   *(mat[3].p+ i*mat[3].c+j)=s;
                       }
           }
```

```
 }
        else throw(mat[1].c);
}
   catch (...)
   {
        cout<<"\ndimensions are not same,so  subtraction not possible\n";
        mat[3].r=0;
        mat[3].c=0;
        return; }
 }
void print(int n,matrix mat[],string m)
{
        for(int e=0;e<n;e++)
        {
                if (e==0)
                cout<<m<<" ,matrix X is: "<<endl;
                else if(e==1)
                cout<<m<<" ,matrix Y is: \n";
                else if(e==2)
                cout<<m<<" ,Z=X+Y matrix: \n";
                else if(e==3)
                cout<<m<<" ,Z=X-Y matrix: \n";
        for(int i=0;i<mat[e].r;i++)
        {
                cout<<"[ ";
                for(int j=0;j<mat[e].c;j++)
                {
                        cout<<*(mat[e].p+i*mat[e].c+j)<<" ";
                }
                cout<<" ]\n";
        }
}
}
```