Aşağıda prototipi verilmiş olan Dizi sınıfı bir veya daha fazla tamsayıyı manipüle etmek için geliştirilmiştir. Dizi maksimum 500 eleman alabilmektedir. Dizinin fonksiyonları ile alakalı yorumlar aşağıda yapılmıştır. Bu yorumlara göre sınıfa ait fonksiyonların gövdelerini sınıfın dışarısında yazın.

```
class Dizi
{
public:
	//Varsayılan kurucu fonksiyon bütün elemanlara 0 atar.
			Dizi();
	//Aşağıdaki kurucu fonksiyon parametre olarak aldığı diziyi
	//kendi içerisine kopyalayacaktır.
			Dizi(int Dizi[],    //Kopyalanacak olan dizi
			     int Count);    //Kopyalanacak dizinin boyutu
	//Aşağıdaki fonksiyon Diziye başka bir dizi nesnesinin kopyalanmasını sağlamaktadır.
			Dizi(Dizi &d);
	//Aşağıdaki fonksiyon dizi içerisinde istenen noktaya başka bir Dizi
	//nesnesi elemanlarının eklenmesini sağlamaktadır.
	//Örnek index=5 ise elemanlar ilk 5 elemandan sonra eklenecek demektir.
	//index -1 girildiğinde veya hiç bir parametre girilmediğinde veriler Dizinin
	//sonuna eklenecektir.
	void   Insert(Dizi& D,            //Elemanları eklenecek olan Dizi nesnesi
			int index=-1);      //kaçıncı elemanından sonra eklenecektir.
	//Aşağıdaki fonksiyon dizinin sonuna yeni bir eleman eklemek için kullanılmaktadır.
	void   Append(int Elem);
	//Aşağıdaki fonksiyon dizinin istenen elemanını geri döndürmektedir.
	int          GetAt(int index);
	//aşağıdaki fonksiyon dizinin istenilen sıradaki elemanına değer atamaktadır.
	void   SetAt(int Elem,int index);
	//aşağıdaki fonksiyon dizinin eleman sayısını döndürmektedir.
	int          GetCount();

protected:
	int m_Elems[500]; //sayıların saklanacağı dizi
	int m_iCounts;    //dizi içerisinde bulunan toplam eleman sayısı
};
```

Uyarı!!!!!(Dizi elemanlarına atama yaparken veya onlara erişirken asla dizi sınırları aşılmamalıdır. fonksiyon gövdeleri tasarlanırken bu husus göz önünde bulundurulmalı ve gerekli önlemler alınmalıdır.)

CEVAP

```
Dizi::Dizi()
{
	for(int i=0;i<500;i++)
		m_Elems[i] = 0 ;

	m_iCounts = 0;
}
Dizi::Dizi(int Dizi[], int Count)
{
	if((m_iCounts+Count)>=500)
		return;
	for(int i=0;i<Count;i++)
		m_Elems[i] = Dizi[i];

	m_iCounts=Count;
}
Dizi::Dizi(Dizi &d)
{
	for(int i=0;i<d.GetCount();i++)
		m_Elems[i] = d.GetAt(i);
	m_iCounts=d.GetCount();
}
int Dizi::GetCount()
{
	return m_iCounts;
}
int Dizi::GetAt(int index)
{
	if(index>500)
		return -1;

	return m_Elems[index];
}
void Dizi::SetAt(int Elem,int index)
{
	if(index>500)
		return ;
	m_Elems[index] = Elem;
}
void Dizi::insert(Dizi &D,int index)
{
	if((m_iCounts+D.GetCount())>=500)
		return;
	if(index==-1)
	{
		for(int i=0;i<D.GetCount();i++)
			m_Elems[m_iCounts+i]=D.GetAt(i);
	}
	else
	{
		for(int i=m_iCounts+D.GetCount()-1;i>=index+D.GetCount();i--)
		{
			m_Elems[i]=m_Elems[i-D.GetCount()];
		}
		for(int i=0;i<D.GetCount();i++)
			m_Elems[i+index] = D.GetAt(i);
	}
	m_iCounts+=D.GetCount();

}
void Dizi::Append(int Elem)
{
	if(m_iCounts+1>=500)
		return;
	m_Elems[m_iCounts] = Elem;

	m_iCounts++;
}
```

Aşağıda 4x4'lük bir Matris sınıfının prototipi görünmektedir.Prototipteki fonksiyonların hemen üstünde yazılmış olan açıklamalara göre her bir fonksiyonun gövdesini sınıf dışarısında yazın

```
class Matrix
{
public:
	//Aşağıdaki fonksiyon matrisin bütün elemanları 0 a eşitler
	Matrix();

	//Aşağıdaki fonksiyon Matrisin elemanlarına dışarıdan girilecek olan 4x4'lük 2
	//boyutlu veya 16 elemanlı tek boyutlu bir diziyi atamaktadır. Gövde
	//içerisinde r parametresinin 16 elemanlı bir dizi olduğu düşünülecektir.
	Matrix(float r[]);

	//Matrisin elemanlarına, parametre olarak girilen başka bir matrisin
	//elemanlarını atamaktadır.
	Matrix(Matrix& Mtx);

	//Aşağıdaki 4 fonksiyon, onları çağıran matris üzerinde işlem yapmamaktadır.
	//İşlemler geçici bir matris sınıfı üzerinde yapılmaktadır.
	//elde edilen matris ise geri döndürülmektedir.
	//Örnek  A = B.Topla(C);  B topla fonksiyonunu çağırmaktadır. işlem sonucunda
	// A matrisi B ile C matrisinin toplam değerine sahip olurken B ile C hiç değişmeyecektir.
	Matrix Topla(Matrix &Mtx);
	Matrix Cikar(Matrix &Mtx);
	Matrix Carp(Matrix &Carp);
	Matrix Bol(float bol);

	//Parametre olarak girilen matrisin elemanları
	//fonksiyonu çağıran matrisin elemanlarına atar.
	void   Ata(Matrix& Mtx);
	//Matrisin her bir satirini kendi içinde küçükten büyüğe soldan sağa sıralayacaktır.
	void   SatirSirala();
	//Matrisin her bir sütununu kendi içinde büyükten küçüğe doğru soldan sağa sıralacaktır.
	void   SutunSirala();
	//Matrisin bütün elemanlarını (0-15 ) siralamaktadır.
	void   TumuSirala();

	//4x4 matrisin elemanlarını tutacaktır.
	float m[4][4];

};
```

CEVAP

```
Matrix::Matrix()
{
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			m[i][j] = 0;

}
Matrix::Matrix(float r[])
{
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			m[i][j] = r[i*4+j];

}

Matrix::Matrix(Matrix& mat)
{
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			m[i][j] = mat.m[i][j];
}
```

```
Matrix Matrix::Cikar(Matrix &mat)
{
	Matrix M;
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			M.m[i][j] =m[i][j]-mat.m[i][j];
	return M;
}
Matrix Matrix::Topla(Matrix &mat)
{
	Matrix M;
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			M.m[i][j] =m[i][j]+mat.m[i][j];
	return M;
}
Matrix Matrix::Bol(float Sayi)
{
	Matrix M;
	for(int i=0;i<4;i++)
		for(int j=0;j<4;j++)
			M.m[i][j] =m[i][j]/Sayi;
	return M;
}
Matrix Matrix::Carp(Matrix &mat)
{
	Matrix M;
	for(int i = 0;i<4;i++)
	{
		for(int j=0;j<4;j++)
		{
			for(int k = 0;k<4;k++)
			{
				M.m[i][j] = m[i][k]*mat.m[k][j];
			}
		}
	}
	return M;
}
void Matrix::SatirSirala()
{
	for(int i=0;i<4;i++)
	{
		bool  temiz = true;

		do
		{
			temiz = true;

			for(int j = 0;j<3;j++)
			{
				if(m[i][j]>m[i][j+1])
				{
					float temp  = m[i][j+1];
					m[i][j+1]   = m[i][j];
					m[i][j]           = temp;
					temiz       = false;
				}
			}

		}while(!temiz);
	}
}
```

```
void Matrix::SutunSirala()
{
	for(int i=0;i<4;i++)
	{
		bool  temiz = true;

		do
		{
			temiz = true;

			for(int j = 0;j<3;j++)
			{
				if(m[j][i]>m[j+1][i])
				{
					float temp  = m[j+1][i];
					m[j+1][i]   = m[j][i];
					m[j][i]           = temp;
					temiz       = false;
				}
			}

		}while(!temiz);
	}
}

void Matrix::TumuSirala()
{
	bool temiz = true;

	do
	{
		temiz = true;
		for(int k = 0;k<15;k++)
		{
			int   i1 = k/4;
			int   j1 = k%4;

			int i2 = (k+1)/4;
			int j2 = (k+1)%4;

			if(m[i1][j1]>m[i2][j2])
			{
				float temp  = m[i2][j2];
				m[i2][j2]   = m[i1][j1];
				m[i1][j1]   = temp;
				temiz       = false;
			}
		}

	}while(!temiz);
}
```