

Chương 10

MẢNG MỘT CHIỀU

CĂN BẢN

cuu duong than cong . com

GV. Nguyễn Sơn Hoàng Quốc
ThS. Nguyễn Tấn Trần Minh Khang

Chương 10- 1

1. VÍ DỤ DẪN NHẬP 1

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng
 - + Nhập mảng một chiều các số nguyên
 - + Xuất mảng một chiều các số nguyên
 - + Tính tổng các giá trị trong mảng
- Chương trình

1. VÍ DỤ DẪN NHẬP 1

```
11. #include <iostream.h>
12. class CMangNguyen
13. {
14.     private:
15.         int a[100];
16.         int n;
17.     public:
18.         void Nhap( );
19.         void Xuat( );
20.         int Tong( );
21. } ;
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void main( )
12. {
13.     CMangNguyen x;
14.     x.Nhap( ) ;
15.     x.Xuat( ) ;
16.     int kq = x.Tong( ) ;
17.     cout << "Tong = " << kq;
18. }
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void CMangNguyen::Nhap( )
12. {
13.     cout<<"Nhap n : ";
14.     cin>>n;
15.     for (int i=0 ; i<n ; i++)
16.     {
17.         cout<<"a["<< i <<"]:";
18.         cin >> a[i];
19.     }
20. }
```

1. VÍ DỤ DẪN NHẬP 1

```
11. void CMangNguyen::Xuat ( )  
12. {  
13.     for (int i=0; i<n ;i++)  
14.         cout << a[i] << "  ";  
15. }
```

cuu duong than cong . com

1. VÍ DỤ DẪN NHẬP 1

```
11. int CMangNguyen::Tong( )
12. {
13.     int s = 0;
14.     for (int i=0;i<n;i++)
15.         s = s + a[i];
16.     return s;
17. }
```

cuu duong than cong . com

2. VÍ DỤ DẪN NHẬP 2

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng
 - + Nhập mảng một chiều các số thực
 - + Xuất mảng một chiều các số thực
 - + Tìm phần tử lớn nhất trong mảng
- Chương trình

2. VÍ DỤ DẪN NHẬP 2

```
11. #include <iostream.h>
12. class CMangThuc
13. {
14.     private:
15.         float a[100];
16.         int n;
17.     public:
18.         void Nhap( );
19.         void Xuat( );
20.         float LonNhat( );
21. } ;
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void main( )
12. {
13.     CMangThuc x;
14.     x.Nhap( ) ;
15.     x.Xuat( ) ;
16.     float kq = x.LonNhat( ) ;
17.     cout << "\n..." << kq;
18. }
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void CMangThuc::Nhap( )
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i=0 ; i<n ; i++)
16.     {
17.         cout << "a[" << i << "] : ";
18.         cin >> a[i];
19.     }
20. }
```

2. VÍ DỤ DẪN NHẬP 2

```
11. void CMangThuc::Xuat( )  
12. {  
13.     for (int i=0 ; i<n ; i++)  
14.         cout << a[i] << " ";  
15. }
```

cuu duong than cong . com

2. VÍ DỤ DẪN NHẬP 2

```
11. float CMangThuc::LonNhat( )
12. {
13.     float lc = a[0];
14.     for (int i=0;i<n;i++)
15.         if (a[i] > lc)
16.             lc = a[i];
17.     return lc;
18. }
```

3. VÍ DỤ DẪN NHẬP 3

- Bài toán: Viết chương trình thực hiện các yêu cầu sau bằng phương pháp lập trình hướng đối tượng:
 - + Nhập mảng một chiều các phân số
 - + Xuất mảng một chiều các phân số
 - + Đếm số lượng giá trị dương có trong mảng
- Chương trình

3. VÍ DỤ DẪN NHẬP 3

```
11. #include <iostream.h>
12. class CPhanSo
13. {
14.     private:
15.         int tu;
16.         int mau;
17.     public:
18.         void Nhap( );
19.         void Xuat( );
20.         int KtDuong( );
21. } ;
```

3. VÍ DỤ DẪN NHẬP 3

```
11. class CMangPhanSo
12. {
13.     private:
14.         CPhanSo a[100];
15.         int n;
16.     public:
17.         void Nhap( );
18.         void Xuat( );
19.         int DemDuong( );
20. } ;
```


3. VÍ DỤ DẪN NHẬP 3

```
11. void main( )
12. {
13.     CMangPhanSo x;
14.     x.Nhap( ) ;
15.     x.Xuat( ) ;
16.     int kq = x.DemDuong( ) ;
17.     cout << "\n..." << kq;
18. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CPhanSo::Nhap( )
12. {
13.     cout<<"Nhap tu : " ;
14.     cin>>tu;
15.     cout<<"Nhap mau : " ;
16.     cin>>mau;
17. }
```

cuu duong than cong . com

3. VÍ DỤ DẪN NHẬP 3

```
11. void CPhanSo::Xuat( )  
12. {  
13.     cout << tu << "/" << mau;  
14. }
```

cuu duong than cong . com

cuu duong than cong . com

3. VÍ DỤ DẪN NHẬP 3

```
11. int CPhanSo::KtDuong( )  
12. {  
13.     if (tu*mau > 0)  
14.         return 1;  
15.     return 0;  
16. }
```

cuu duong than cong . com

3. VÍ DỤ DẪN NHẬP 3

```
11. void CMangPhanSo::Nhap( )
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i=0;i<n;i++)
16.     {
17.         cout<<"Nhap a["<<i<<"]:";
18.         a[i].Nhap( );
19.     }
20. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. void CMangPhanSo::Xuat ( )
12. {
13.     for (int i=0;i<n;i++)
14.     {
15.         a[i].Xuat ( ) ;
16.         cout << " ";
17.     }
18. }
```

3. VÍ DỤ DẪN NHẬP 3

```
11. int CMangPhanSo::DemDuong( )
12. {
13.     int dem = 0;
14.     for (int i=0;i<n;i++)
15.         if (a[i].KtDuong( )==1)
16.             dem = dem + 1;
17.     return dem;
18. }
```

cuu duong than cong . com

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

- Hãy xây dựng lớp số nguyên với các phương thức như sau:
 - + Phương thức nhập mảng
 - + Phương thức xuất mảng
 - + Phương thức liệt kê các giá trị chứa trong mảng
 - + Phương thức tính tổng các phần tử trong mảng
 - + Phương thức tính tổng các giá trị cực đại trong mảng
 - + Phương thức đếm số lượng giá trị lẻ có trong mảng
 - + Phương thức đếm số lần xuất hiện của giá trị x trong mảng

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

- Hãy xây dựng lớp số nguyên với các phương thức như sau:
 - + Phương thức kiểm tra mảng có tồn tại giá trị 0 hay không?
 - + Phương thức kiểm tra mảng có toàn chẵn hay không?
 - + Phương thức kiểm tra mảng có tăng dần hay không?
 - + Phương thức sắp xếp các giá trị trong mảng tăng dần
 - + Phương thức sắp xếp các giá trị trong mảng giảm dần
 - + Phương thức sắp xếp các giá trị lẻ trong mảng tăng dần

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. class CMangNguyen
12. {
13.     private:
14.         int a[100];
15.         int n;
16.     public:
17.         void Nhap();
18.         void Xuat();
19.         void LietKeChan();
20.         int TinhTong();
21.         int LonNhat();
22.         int TongCucDai();
23.         int DemLe();
24.         int TanSuat(int);
25.         int KTTonTaiKhong();
26.         int KTToanChan();
27.         int KTTangDan();
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. class CMangNguyen
12. {
13.     private:
14.         int a[100];
15.         int n;
16.     public:
17.         void SapTang( ) ;
18.         void SapGiam( ) ;
19.         void SapLeTang( ) ;
20. } ;
```

cuu duong than cong . com

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::Nhap( )
12. {
13.     cout<<"Nhap n : ";
14.     cin>>n;
15.     for (int i=0 ; i<n ; i++)
16.     {
17.         cout<<"a[ "<<i<< " ]:";
18.         cin >> a[i];
19.     }
20. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::Xuat( )  
12. {  
13.     for (int i=0;i<n;i++)  
14.         cout <<a[i]<<"    ";  
15. }
```

cuu duong than cong . com

cuu duong than cong . com

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::LietKeChan( )
12. {
13.     for (int i=0; i<n; i++)
14.         if (a[i]%2 == 0)
15.             cout << a[i] << " ";
16. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::TinhTong( )
12. {
13.     int s = 0;
14.     for (int i=0;i<n;i++)
15.         s = s + a[i];
16.     return s;
17. }
```

cuu duong than cong . com

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::LonNhat ( )
12. {
13.     int lc = a[0];
14.     for (int i=0;i<n;i++)
15.         if (a[i] > lc)
16.             lc = a[i];
17.     return lc;
18. }
```


4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::TongCucDai ( )
12. {
13.     if (n <= 1)
14.         return 0;
15.     int s = 0;
16.     if (a[0] > a[1])
17.         s = s + a[0];
18.     for (int i = 1; i <= n - 2; i++)
19.         if (a[i] > a[i - 1] &&
20.             a[i] > a[i + 1])
21.             s = s + a[i];
22.     if (a[n - 1] > a[n - 2])
23.         s = s + a[n - 1];
24.     return s;
25. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::DemLe( )
12. {
13.     int dem = 0;
14.     for (int i=0; i<n; i++)
15.         if (a[i]%2!=0)
16.             dem = dem + 1;
17.     return dem;
18. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::TanSuat(int x)
12. {
13.     int dem = 0;
14.     for (int i=0; i<n; i++)
15.         if (a[i]==x)
16.             dem = dem + 1;
17.     return dem;
18. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::KTTonTaiKhong()  
12. {  
13.     int flag = 0;  
14.     for (int i=0; i<n; i++)  
15.         if (a[i]==0)  
16.             flag = 1;  
17.     return flag;  
18. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::KTToanChan( )
12. {
13.     int flag = 1;
14.     for (int i=0; i<n; i++)
15.         if (a[i]%2!=0)
16.             flag = 0;
17.     return flag;
18. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. int CMangNguyen::KTTangDan( )
12. {
13.     int flag = 1;
14.     for (int i=0;i<=n-2;i++)
15.         if (a[i]>a[i+1])
16.             flag = 0;
17.     return flag;
18. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::SapTang( )
12. {
13.     for (int i=0;i<=n-2;i++)
14.         for (int j=i+1;j<=n-1;j++)
15.             if (a[i]>a[j])
16.                 {
17.                     int temp = a[i];
18.                     a[i] = a[j];
19.                     a[j] = temp;
20.                 }
21. }
```

4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::SapGiam( )
12. {
13.     for (int i=0;i<=n-2;i++)
14.         for (int j=i+1;j<=n-1;j++)
15.             if (a[i]<a[j])
16.                 {
17.                     int temp = a[i];
18.                     a[i] = a[j];
19.                     a[j] = temp;
20.                 }
21. }
```


4. XÂY DỰNG LỚP MẢNG SỐ NGUYÊN

```
11. void CMangNguyen::SapLeTang( )
12. {
13.     for (int i=0;i<=n-2;i++)
14.         for (int j=i+1;j<=n-1;j++)
15.             if(a[i] < a[j] &&
16.                 a[i]%2!=0 &&
17.                 a[j]%2!=0)
18.                 {
19.                     int temp = a[i];
20.                     a[i] = a[j];
21.                     a[j] = temp;
22.                 }
23. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

- Hãy xây dựng lớp số thực với các phương thức như sau:
 - + Phương thức nhập mảng
 - + Phương thức xuất mảng
 - + Phương thức tính tổng các phần tử trong mảng
 - + Phương thức tìm phần tử nhỏ nhất trong mảng
 - + Phương thức tính tổng các giá trị cực tiểu trong mảng
 - + Phương thức đếm số lần xuất hiện của giá trị x trong mảng

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

- Hãy xây dựng lớp số thực với các phương thức như sau:
 - + Phương thức kiểm tra mảng có tồn tại giá trị 0 hay không?
 - + Phương thức kiểm tra mảng có tăng dần hay không?
 - + Phương thức sắp xếp các giá trị trong mảng tăng dần
 - + Phương thức sắp xếp các giá trị trong mảng giảm dần

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. class CMangThuc
12. {
13.     private:
14.         float a[100];
15.         int n;
16.     public:
17.         voidNhap();
18.         voidXuat();
19.         float TinhTong();
20.         float NhoNhat();
21.         float TongCucTieu();
22.         int DemXuatHien(float);
23.         int KTTonTaiKhong();
24.         int KTTangDan();
25.         void SapTang();
26.         void SapGiam();
27. };
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. void CMangThuc::Nhap( )
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i=0 ; i<n ; i++)
16.     {
17.         cout<<"a["<<i<< "]: ";
18.         cin >> a[i];
19.     }
20. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. void CMangThuc::Xuat( )
12. {
13.     for (int i=0;i<n;i++)
14.         cout<<a[i]<<" ";
15. }
```

cuu duong than cong . com

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. float CMangThuc::TinhTong( )
12. {
13.     float s = 0;
14.     for (int i=0; i<n; i++)
15.         s = s + a[i];
16.     return s;
17. }
```

cuu duong than cong . com

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. float CMangThuc::NhoNhat ( )
12. {
13.     float lc = a[0];
14.     for (int i=0;i<n;i++)
15.         if (a[i]<lc)
16.             lc = a[i];
17.     return lc;
18. }
```


5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. int CMangThuc::TanSuat(float x)
12. {
13.     int dem = 0;
14.     for (int i=0; i<n; i++)
15.         if (a[i]==x)
16.             dem = dem + 1;
17.     return dem;
18. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. int CMangThuc::KTTonTaiKhong( )
12. {
13.     for (int i=0;i<n;i++)
14.         if (a[i]==0)
15.             return 1;
16.     return 0;
17. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. int CMangThuc::KTTangDan( )
12. {
13.     for (int i=1; i<n; i++)
14.         if (a[i-1]<a[i])
15.             return 1;
16.     return 0;
17. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. void CMangThuc::SapTang( )
12. {
13.     for(int i=0;i<=n-2;i++)
14.         for(int j=i+1;j<=n-1;j++)
15.             if(a[i]>a[j])
16.                 {
17.                     float temp = a[i];
18.                     a[i] = a[j];
19.                     a[j] = temp;
20.                 }
21. }
```

5. XÂY DỰNG LỚP MẢNG SỐ THỰC

```
11. void CMangThuc::SapGiam( )
12. {
13.     for(int i=0;i<=n-2;i++)
14.         for(int j=i+1;j<=n-1;j++)
15.             if(a[i]>a[j])
16.                 {
17.                     float temp = a[i];
18.                     a[i] = a[j];
19.                     a[j] = temp;
20.                 }
21. }
```