

BUỔI THỰC HÀNH 1: ÔN TẬP CÁC THUẬT TOÁN SẮP XẾP CƠ BẢN

1.1. Bài thực hành 1: Viết chương trình nhập danh sách đặc số nguyên a gồm n phần tử ($0 < n \leq 100$). Hãy xuất danh sách đặc a vừa nhập

Yêu cầu xây dựng các hàm:

- Nhập danh sách đặc a
- Xuất danh sách đặc a.

Hướng dẫn:

```
void nhapmang(int a[], int &n)
{
    do{
        cout<<" So luong phan tu mang:";
        cin>>n;
    }while(n <=0 || n > 100);
    for(int i = 0; i < n; i++)
    {
        cout<<"a[<<i<<"]="<<endl;
        cin>>a[i];
    }
}

void xuatmang(int a[],int n)
{
    cout<<"Mang a:";
```

```

        for( int i= 0; i < n ; i++)
            cout<<"a["<<i<<"]="<<a[i]<<endl;
    }

    void main()
    {
        int a[100], n;
        nhapmang(a,n);
        xuatmang(a, n);
        system("pause");
    }

```

1.2. Bài thực hành 2: Quản lý danh sách đặc 100 phân tử kiểu số nguyên (int)

1.2.1. Khai báo cấu trúc danh sách.

1.2.2. Viết thủ tục nhập danh sách.

1.2.3. Viết thủ tục xuất danh sách

1.2.4. Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán

Selection Sort

Hướng dẫn:

```

void nhapmang(int a[], int &n)
{
    do{
        cout<<"\n So luong phan tu mang:";
        cin>>n;
    }while(n <=0 || n > 100);
    for(int i = 0; i < n; i++)

```

```

        {
            cout<<"\na["<<i<<"]="";
            cin>>a[i];
        }
    }

```

```

void hoanvi(int &x, int &y)
{
    int tam = x;
    x = y;
    y = tam;
}

```

```

void SelectionSort(int a[], int n )
{
    int min;
    for (int i=0; i<n-1 ; i++)
    {
        min = i;
        for(int j = i+1; j < n ; j++)
            if (a[j] < a[min])
                min = j;
        if (min != i)
            hoanvi(a[min], a[i]);
    }
}

```

```

void xuatmang(int a[], int n)
{
    cout<<"Mang a sau khi sap tang:\n";
    for(int i = 0; i < n; i++)
        cout<<a[i];
}

void main()
{
    int a[100], n;
    nhapmang(a, n);
    SelectionSort(a, n);
    xuatmang(a, n);
    system("pause");
}

```

1.3. Bài thực hành 3: Quản lý danh sách đặc 100 phần tử kiểu số nguyên (int)

1.3.1. Khai báo cấu trúc danh sách.

1.3.2. Viết thủ tục nhập danh sách.

1.3.3. Viết thủ tục xuất danh sách

1.3.4. Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán

Interchange Sort

Hướng dẫn:

```

void nhapmang(int a[], int &n)
{
    do{

```

```

        cout<<"\n So luong phan tu mang:";
        cin>>n;
    }while(n <=0 || n > 100);
    for(int i = 0; i < n; i++)
    {
        cout<<"\na["<<i<<"]="";
        cin>>a[i];
    }
}

```

```

void hoanvi(int &x, int &y)
{
    int tam = x;
    x = y;
    y = tam;
}

```

```

void InterchangeSort(int a[], int n)
{
    int i, j;
    for (i = 0 ; i<n-1 ; i++)
        for (j =i+1; j < n ; j++)
            if(a[j]< a[i])
                hoanvi(a[i], a[j]);
}

```

```

void xuatmang(int a[], int n)

```

```

    {
        cout<<"Mang a sau khi sap tang:\n";
        for(int i = 0; i < n; i ++)
            cout<<a[i];
    }

void main()
{
    int a[100], n;
    nhapmang(a, n);
    InterchangeSort (a, n);
    xuatmang(a, n);
    system("pause");
}

```

1.4. Bài thực hành 4: Quản lý danh sách đặc 100 phần tử kiểu số nguyên (int)

1.4.1. Khai báo cấu trúc danh sách.

1.4.2. Viết thủ tục nhập danh sách.

1.4.3. Viết thủ tục xuất danh sách

1.4.4. Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán Bubble Sort

Hướng dẫn:

```

void nhapmang(int a[], int &n)
{
    do{
        cout<<"\n So luong phan tu mang:";

```

```

        cin>>n;
    }while(n <=0 || n > 100);
    for(int i = 0; i < n; i++)
    {
        cout<<"\na["<<i<<"]=";
        cin>>a[i];
    }
}

```

```

void hoanvi(int &x, int &y)
{
    int tam = x;
    x = y;
    y = tam;
}

```

```

void BubbleSort(int a[], int n)
{
    int i, j;
    for (i = 0 ; i<n-1 ; i++)
        for (j =n-1; j>i ; j --)
            if(a[j]< a[j-1])
                hoanvi(a[j], a[j-1]);
}

```

```

void xuatmang(int a[], int n)
{

```

```
        cout<<"Mang a sau khi sap tang:\n";  
        for(int i = 0; i < n; i ++)  
            cout<<a[i];  
    }
```

```
void main()  
{  
    int a[100], n;  
    nhapmang(a, n);  
    BubbleSort (a, n);  
    xuatmang(a, n);  
    system("pause");  
}
```