**ĐỀ THI TỐT NGHIỆP NĂM 2020 (LẦN 1)**

**Câu 1**: **(3.5 điểm)**

Sử dụng ngôn ngữ lập trình C#, hãy cài đặt các phương thức sau để thực hiện các thao tác trên mảng 1 chiều, có n phần tử lưu trữ giá trị kiểu số nguyên.

1. Phương thức nhập mảng từ bàn phím

**public static void NhapMang(int[] a, int n) { ... }**

static void NhapMang(int[] a,int n)

{

for(int i=0;i<n;i++)

{

Console.Write("Nhap a[{0}]", i);

a[i] = int.Parse(Console.ReadLine());

}

}

1. Phương thức xuất mảng ra màn hình

**public static void XuatMang(int[] a, int n) { ... }**

static void XuatMang(int[] a, int n)

{

for(int i=0; i<n; i++)

{

Console.Write("{0} ", a[i]);

}

}

1. Phương thức tìm phần tử có giá trị lớn nhất trong mảng

**public static int TimMax(int[] a, int n) { ... }**

static int TimMax(int[] a, int n)

{

int max = a[0];

for (int i=1; i<n; i++)

{

if(max<a[i])

{

max = a[i];

}

}

return max;

}

1. Sắp xếp mảng tăng dần bằng thuật toán đổi chỗ trực tiếp (InterChangeSort).

**public static void SapXepTang(int[] a, int n) { ... }**

static void SapXepTang(int[] a, int n)

{

int tam;

for(int i=0;i<n-1;i++)

{

for(int j=i+1; j<n;j++)

{

if(a[i]>a[j])

{

tam = a[i];

a[i] = a[j];

a[j] = tam;

}

}

}

}

Áp dụng thuật toán trên với bộ dữ liệu K = { 40, 24, 18, 53, 20, 11}

Hãy chỉ rõ kết quả từng bước thực hiện của thuật toán để sắp xếp mảng tăng dần.

|  |  |
| --- | --- |
| Bước i | Kết quả |
| i=0 | 11 40 24 53 20 18 |
| i=1 | 11 18 40 53 24 20 |
| i=2 | 11 18 20 53 40 24 |
| i=3 | 11 18 20 24 53 40 |
| i=4 | 11 18 20 24 40 53 |

1. Sắp xếp mảng tăng dần bằng thuật toán chọn trực tiếp (SelectionSort).

**public static void SapXepTang(int[] a, int n) { ... }**

static void SapXepTang(int[] a, int n)

{

int tam;

for(int i=0;i<n-1;i++)

{

int min = i;

for(int j=i+1; j<n;j++)

{

if(a[min]>a[j])

{

min = j;

}

}

if(min!=i)

{

tam = a[i];

a[i] = a[min];

a[min] = tam;

}

}

}

Áp dụng thuật toán trên với bộ dữ liệu K = { 40, 24, 18, 53, 20, 11}

Hãy chỉ rõ kết quả từng bước thực hiện của thuật toán để sắp xếp mảng tăng dần.

|  |  |
| --- | --- |
| Bước i | Kết quả |
| i=0 | 11 24 18 53 20 40 |
| i=1 | 11 18 24 53 20 40 |
| i=2 | 11 18 20 53 24 40 |
| i=3 | 11 18 20 24 53 40 |
| i=4 | 11 18 20 24 40 53 |

**Câu 2**: **(1.5 điểm)**

Để biểu diễn cấu trúc dữ liệu danh sách liên kết (DSLK) đơn lưu trữ giá trị (Info) kiểu số nguyên, ta định nghĩa các lớp sau bằng ngôn ngữ lập trình C#:

|  |  |
| --- | --- |
| //Lớp định nghĩa cấu trúc 1 nút  class Node {  public int Info { set; get; }  public Node Next { set; get; }  public Node(int x) {  Info = x;  Next = null;  }  } | //Lớp định nghĩa cấu trúc danh //sách liên kết đơn:  class LinkedList {  public Node Head { set; get; }  public LinkedList()  {  Head = null;  }  } |

Sử dụng ngôn ngữ C#, Anh(chị) hãy cài đặt các phương thức trong lớp LinkedList để thực hiện các yêu cầu sau:

1. Thêm một nút có thông tin là X vào đầu danh sách.

**public void themDau (int X) { …. }**

public void themDau(int X)

{

Node p = new Node(X);

if(Head==null)

{

Head = p;

}else

{

p.Next = Head;

Head = p;

}

}

1. Đếm có bao nhiêu nút trong danh sách

**public int demNut()** {

int dem =0;

Node p = Head;

while(p!=null)

{

dem++;

p = p.Next;

}

return dem;

}

1. Tính giá trị trung bình các nút có giá trị dương chẳn trong danh sách

**public double tinhTBDuongChan() { …. }**

public double tinhTBDuongChan()

{

int tong = 0;

int dem = 0;

Node p = Head;

while(p!=null)

{

if (p.Info > 0 && p.Info % 2 == 0)

{

tong += p.Info;

dem++;

}

p = p.Next;

}

return (double)tong / dem;

}