**// De: Sap xep chon, tang dan**

#include <iostream>

#include <fstream>

using namespace std;

//Khai bao ham

void SelectSort(int \*a, int n);

//===chuong trinh chinh===

int main()

{

ifstream fin("daykhoa.txt");

int \*a;

int n, i;

//Doc vao so phan tu cua day khoa

fin>>n;

//Tao mang dong chua n o nho

a = new int[n];

//Doc vao day khoa

for(i=0; i<n; i++) fin>>a[i];

//Dua ra day khoa ban dau

cout<<"Day khoa ban dau doc duoc tu tep la:\n";

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

//Sap xep

SelectSort(a, n);

//Dua ra day khoa da sap xep

cout<<"\n\nDay khoa sau khi sap xep la:\n";

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

cout<<endl; return 0;

}

//===dinh nghia ham===

void SelectSort(int \*a, int n)

{

int i,j,k,tg;

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

{

//Tim vi tri k nho nhat

//Giam dan: Tim vi tri k lon nhat

k = i;

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

if(a[j] < a[k]) k = j;

//Giam dan: if(a[j] > a[k]) k = j;

//Doi cho vi phan tu tai tri nho nhat voi phan tu vi tri i

if(k != i)

{

tg = a[i];

a[i] = a[k];

a[k] = tg;

}

}

}

**// De: Sap xep chen, tang dan**

#include <iostream>

#include <fstream>

using namespace std;

//Khai bao ham

void InsertSort(int \*a, int n);

//===chuong trinh chinh===

int main()

{

ifstream fin("daykhoa.txt");

int \*a;

int n, i;

//Doc vao so phan tu cua day khoa

fin>>n;

//Tao mang dong chua n phan tu

a = new int[n];

//Doc vao day khoa

for(i=0; i<n; i++) fin>>a[i];

//Dua ra day khoa ban dau

cout<<"Day khoa ban dau la:\n";

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

//Sap xep

InsertSort(a, n);

//Dua ra day khoa da sap xep

cout<<"\n\nDay khoa sau khi sap xep la:\n";

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

cout<<endl;

return 0;

}

//===dinh nghia ham===

void InsertSort(int \*a, int n)

{

int i,j,tg;

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

{

tg = a[i]; j = i-1;

//Tim vi tri thich hop de chen tg

while(tg<a[j] && j>-1) a[j+1] = a[j--];

//giam dan: while(tg>a[j] && j>-1) a[j+1] = a[j--];

//Chen tg vao vi tri da tim duoc

a[j+1] = tg;

}

}

**//De: Sap xep nhanh**

#include<iostream>

#include<fstream>

using namespace std;

//Khai bao ham

void QuickSort(int\* &a,int L ,int R);

//===chuong trinh chinh===

int main()

{

int \*a;

int n,i;

ifstream fin("daykhoa.txt");

//Doc vao so phan tu

fin>>n;

//Tao mang dong a co n o nho

a = new int[n];

//Doc day khoa tu tep vao mang dong a

for(i=0;i<n;i++) fin>>a[i];

//Dua ra

cout<<"Day khoa ban dau doc duoc tu tep la:\n";

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

//Sap xep

QuickSort(a,0,n-1);

//Dua ra

cout<<"\n\nDay khoa sau khi sap xep la:\n";

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

//Xoa mang dong

delete a;

cout<<endl;

return 0;

}

//===dinh nghia ham===

void QuickSort(int\* &a,int L ,int R)

{

if(L>=R) return;

int i=L,j=R,k=(L+R)/2,x=a[k],tg;

//Phan doan

do

{

//Duyet tu trai sang phai cho toi khi gap ai>=x

//Giam dan: Duyet tu trai sang phai cho toi khi gap ai<=x

while(a[i]<x) i++;

//Giam dan: while(a[i]>x) i++;

//Duyet tu phai sang trai cho toi khi gap aj<=x

//Giam dan: Duyet tu phai sang trai cho toi khi gap aj>=x

while(a[j]>x) j--;

//Giam dan: while(a[j]<x) j--;

//Doi cho ai va aj

if(i<j)

{ tg=a[i];

a[i]=a[j];

a[j]=tg;

}

}

while(i<j);

//Phan doan con ben trai

QuickSort(a,L,j-1);

//Phan doan con ben phai

QuickSort(a,j+1,R);

}

**//De: Sap xep sui bot, tang dan**

#include <iostream>

#include <fstream>

using namespace std;

//Khai bao ham

void BubbleSort(int \*a, int n);

//===chuong trinh chinh===

int main()

{

ifstream fin("daykhoa.txt");

int \*a;

int n, i;

//Doc vao so phan tu cua day khoa

fin>>n;

//Tao mang dong chua n phan tu

a = new int[n];

//Doc vao day khoa

for(i=0; i<n; i++) fin>>a[i];

//Dua ra day khoa ban dau

cout<<"Day khoa ban dau doc duoc tu tep la:\n";

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

//Sap xep

BubbleSort(a, n);

//Dua ra day khoa da sap xep

cout<<"\n\nDay khoa sau khi sap xep la:\n";

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

//Xoa mang dong

delete a;

cout<<endl;

return 0;

}

//===dinh nghia ham===

void BubbleSort(int \*a, int n)

{

int i,j,tg;

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

for(j=n-1; j>i; j--)

if(a[j] < a[j-1]) //Giam dan: if(a[j] > a[j-1])

{

tg = a[j];

a[j] = a[j-1];

a[j-1] = tg;

}

}

**//De: Cai dat giai thuat tim kiem nhi phan tren mang ko de quy**

#include<iostream>

#include<stdio.h>

#include<fstream>

using namespace std;

//.Khai bao ham

int BinarySearch(int \*a,int n,int x);

//===chuong trinh chinh===

int main()

{

ifstream fin("daykhoatangdan.txt");

int \*a,n,i,x;

//Doc vao so phan tu cua day khoa

fin>>n;

//Tao mang dong a co n o nho

a=new int[n];

//Doc day khoa tu tep vao mang a

for(i=0;i<n;i++) fin>>a[i];

//Dua ra man hinh

cout<<"Day khoa doc duoc tu tep la:\n";

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

cout<<"\n\nNhap vao gia tri can tim kiem x= ";

cin>>x;

int kq=BinarySearch(a,n,x);

if(kq)

printf("\nTim thay gia tri %d tai vi tri %d trong day.",x,kq);

else

printf("\nKhong tim thay gia tri %d trong day",x);

//Xoa mang dong

delete a;

cout<<endl;

return 0;

}

//===dinh nghia ham===

int BinarySearch(int \*a,int n,int x)

{

//1. Khoi tao

int L=0,R=n-1,m;

//2. Tim kiem

while(L<=R)

{

//3.tinh chi so giua

m=(L+R)/2;

//4.So sanh

if(x<a[m]) R = m-1;

else

if (x>a[m]) L = m+1;

else return m+1;

}

//5. Truong hop khong tim thay

return 0;

}

**//De: Cai dat DSLKK voi phan tu du lieu la so nguyen**

#include<iostream>

#include<stdio.h>

using namespace std;

//Khai bao cau truc Node

struct Node

{

int Infor;

Node \*Left, \*Right;

};

//Khai bao cau truc DSLKK

struct DList

{

Node \*L, \*R;

};

//Khai bao cac ham

//---cac ham bo sung phan tu du lieu x vao DSLKK----

void DLLPreInsert(DList &LR,Node \*M,int x); // bo sung x vao truoc nut M

void DLLPostInsert(DList &LR, Node \*M, int x); // bo sung x vao sau nut M

void DLLFirstInsert(DList &LR, int x); // bo sung x vao dau danh sach

void DLLLastInsert(DList &LR, int x); // bo sung x vao cuoi danh sach

//---cac ham loai bo mot nut khoi DSLKD----

int DLLFirstDelete(DList &LR); // xoa nut dau danh sach

int DLLLastDelete(DList &LR); //xoa nut cuoi danh sach

int DLLDeleteM(DList &LR, Node \*M); // xoa nut M

int DLLDeleteX(DList &LR, int x); // xoa nut co gia tri = x

//Ham duyet danh sach

void DLLDisplay(DList LR,bool DuyetTrai=1);

//===chuong trinh chinh===

int main()

{

//Cài đặt lưu trữ dslkk

DList LR={NULL,NULL};

//---CODE TEST CAC HAM---

//Test cac ham Insert

//Them vao truoc nut M - M la nut dau

DLLPreInsert(LR,LR.L,4);//4

DLLPreInsert(LR,LR.L,3);//3 4

DLLPreInsert(LR,LR.L,1);//1 3 4

//Them vao sau nut M - TH M la nut cuoi

DLLPostInsert(LR,LR.R,5);//1 3 4 5

//Them vao sau nut M - TH M la nut dau

DLLPostInsert(LR,LR.L,2);//1 2 3 4 5

//Them vao dau danh sach

DLLFirstInsert(LR,0);//0 1 2 3 4 5

//Them vao cuoi danh sach

DLLLastInsert(LR,6);//0 1 2 3 4 5 6

//Test các hàm loại bỏ

//Xoa nut dau

DLLFirstDelete(LR);//1 2 3 4 5 6

//Xoa nut cuoi

DLLLastDelete(LR);//1 2 3 4 5

//Xoa nut M - TH M la nut dau

DLLDeleteM(LR,LR.L);//2 3 4 5

//Xoa nut M - TH M la nut cuoi

DLLDeleteM(LR,LR.R);//2 3 4

//Xoa nut M TH con lại (Test thử với nút thứ 2)

DLLDeleteM(LR,LR.L->Right);//2 4

//Xoa nut co gia tri = x - TH nut co gia tri = x la nut dau

DLLDeleteX(LR,2);//4

//Xoa nut cuoi

DLLDeleteX(LR,4); // rong

//... con nua

cout<<"DSLKK duyet tu trai sang phai:\n";

DLLDisplay(LR);

cout<<"\n\nDSLKK duyet tu phai sang trai:\n";

DLLDisplay(LR,0);

//---END CODE TEST----

cout<<endl;

return 0;

}

//===dinh nghia ham===

//---cac ham bo sung phan tu du lieu x vao DSLKK----

void DLLPreInsert(DList &LR,Node \*M,int x)// bo sung x vao truoc nut M

{

//1.Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Left = N->Right = NULL;

//2.Truong hop ds rong

if(LR.L==NULL)

{

LR.L = LR.R = N;

return;

}

//3.Truong hop M la nut cuc trai

if(M==LR.L)

{

N->Right = LR.L;

LR.L->Left = N;

LR.L = N;

return;

}

//4.Truong hop con lai

M->Left->Right = N;

N->Left = M->Left;

N->Right = M;

M->Left = N;

}

//---------------------------

void DLLPostInsert(DList &LR, Node \*M, int x) // bo sung x vao sau nut M

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Left = N->Right = NULL;

//2. TH Danh sach rong

if(LR.L==NULL)

{

LR.L = LR.R = N;

return;

}

//3. TH M la nut cuc phai

if(LR.R==M)

{

M->Right = N;

N->Left = M;

LR.R = N;

return;

}

//4. TH con lai

N->Right = M->Right;

N->Right->Left = N;

N->Left = M;

M->Right = N;

}

//---------------------------

void DLLFirstInsert(DList &LR, int x) // bo sung x vao dau danh sach

{

//1.Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Left = N->Right = NULL;

//2.Truong hop ds rong

if(LR.L==NULL)

{

LR.L=LR.R=N;

return;

}

//3. Bo sung vao nut dau danh sach

N->Right = LR.L;

LR.L->Left = N;

LR.L = N;

}

//---------------------------

void DLLLastInsert(DList &LR, int x) // bo sung x vao cuoi danh sach

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Left = N->Right = NULL;

//2. TH Danh sach rong

if(LR.L==NULL)

{

LR.L = LR.R = N;

return;

}

//3. Bo sung x vao nut cuoi danh sach

LR.R->Right = N;

N->Left = LR.R;

LR.R = N;

}

//---cac ham loai bo mot nut khoi DSLKD----

int DLLFirstDelete(DList &LR) // xoa nut dau danh sach

{

//1. Kiem tra danh sach rong

if(LR.L==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

//2. Giu lai gia tri nut dau

int Tg = LR.L->Infor;

//3. Xoa nut dau

LR.L = LR.L->Right; //Da bao gom ca truong hop chi co 1 nut

LR.L->Left = NULL;

//4. Tra ve gia tri nut dau

return Tg;

}

//---------------------------

int DLLLastDelete(DList &LR) //xoa nut cuoi danh sach

{

//1. Kiem tra danh sach rong

if(LR.L==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

//2. Giu lai gia tri nut cuoi

int Tg = LR.R->Infor;

//3. Xoa nut cuoi

LR.R = LR.R->Left; //Da bao gom ca truong hop chi co 1 nut

LR.R->Right = NULL;

//4. Tra ve gia tri nut dau

return Tg;

}

//---------------------------

int DLLDeleteM(DList &LR, Node \*M) // xoa nut M

{

//1. Kiem tra danh sach rong

if(LR.L==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

//2. Giu lai gia tri nut bi loai bo

int Tg = M->Infor;

//3. Thay doi lien ket

if(M==LR.L && LR.L==LR.R) LR.L = LR.R = NULL;//Truong hop danh sach co 1 nut

else

if(LR.L==M)//Truong hop M la nut cuc trai

{

LR.L = LR.L->Right;

LR.L->Left = NULL;

}

else

if(LR.R==M)//Truong hop M la nut cuc phai

{

LR.R = LR.R->Left;

LR.R->Right = NULL;

}

else//Truong hop con lai

{

M->Left->Right = M->Right;

M->Right->Left = M->Left;

}

//4.Huy nut M

delete M;

//5. Tra ve gia tri cua nut bi loai bo

return Tg;

}

//---------------------------

int DLLDeleteX(DList &LR, int x) // xoa nut co gia tri = x

{

//1. Kiem tra danh sach rong

if(LR.L==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

Node \*P = LR.L;

while(P)//Duyet neu con co nut

{

if(P->Infor == x)//Tim duoc nut x can xoa

{

If(LR.L==LR.R && LR.L==P) LR.L=LR.R=NULL;//TH chi co 1 nut

Else //TH nut can xoa la nut cuc trai

if(LR.L==P)

{

LR.L = LR.L->Right;

LR.L->Left = NULL;

}

else

if(LR.R==P) //TH nut can xoa la nut cuc phai

{

LR.R = LR.R->Left;

LR.R->Right = NULL;

}

else //TH con lai

{

P->Left->Right = P->Right;

P->Right->Left = P->Left;

}

//Xoa nut co gia tri = x

delete P;

//Xoa thanh cong

return 1;

}

//Duyet nut tiep theo (Khong can else vi vao if thi da return roi)

P = P->Right;

}

//Xoa that bai

return 0;

}

//---------------------------

//Ham duyet danh sach

void DLLDisplay(DList LR, bool DuyetTrai)

{

// 2 cách code đều cùng độ phức tạp tính toán O -- cách thày code ngắn gọn hơn nhưng cần tạo biến trung gian

/\*=---code cua thay---

Node \*P;

if(DuyetTrai) P=LR.L;

else P=LR.R;

while(P)

{

cout<<P->Infor<<" ";

if(DuyetTrai) P=P->Right;

else P=P->Left;

}

\*/

//TH duyet tu trai qua phai

if(DuyetTrai)

{

while(LR.L)

{

cout<<LR.L->Infor<<" ";

LR.L = LR.L->Right;

}

return;

}

//TH duyet tu phai qua trai

while(LR.R)

{

cout<<LR.R->Infor<<" ";

LR.R = LR.R->Left;

}

}

**//De: Cai dat cau truc du lieu danh sach lien ket don**

#include <iostream>

#include <fstream>

using namespace std;

//Khai bao cau truc Node

struct Node

{

int Infor;

Node \*Link;

};

//Khai bao cac ham

//---cac ham bo sung phan tu du lieu x vao DSLKD----

void BeforeInsert(Node\* &F, Node \*M, int x); // bo sung x vao truoc M

void PostInsert(Node\* &F, Node \*M, int x); // bo sung x vao sau nut M

void FirstInsert(Node\* &F, int x); // bo sung x vao dau danh sach

void LastInsert(Node\* &F, int x); // bo sung x vao cuoi danh sach

//---cac ham loai bo mot nut khoi DSLKD----

int FirstDelete(Node\* &F); // xoa nut dau danh sach

int LastDelete(Node\* &F); //xoa nut cuoi danh sach

int DeleteM(Node\* &F, Node \*M); // xoa nut M

int DeleteX(Node\* &F, int x); // xoa nut co gia tri = x

//---ham duyet danh sach---

void Display(Node\* F);

//===chuong trinh chinh===

int main()

{

//Cai dat cau truc DSLKD

Node \*F = NULL;

//---CODE TEST CAC HAM---

//Test chen trc M

BeforeInsert(F,F,4); //4

BeforeInsert(F,F,1);//1 4

//Bo sung truoc nut cuoi

//Tìm nút cuối

Node \*Last=F;

while(Last->Link!=NULL) Last=Last->Link;

BeforeInsert(F,Last,3);//1 3 4

//Test chèn sau M

PostInsert(F,F,2); //1 2 3 4

//Tìm nút cuối

Last=F;

while(Last->Link!=NULL) Last=Last->Link;

PostInsert(F,Last,5);//1 2 3 4 5

//Test chèn vào đầu dánh sách

FirstInsert(F,0);//0 1 2 3 4 5

//Test chèn vào cuối danh sách

LastInsert(F,7);// 0 1 2 3 4 5 7

//Test xóa nút đầu

FirstDelete(F);//1 2 3 4 5 7

//Test xóa nút cuối

LastDelete(F);//1 2 3 4 5

//Test xóa nút M

//TH M la nút đầu

DeleteM(F,F);//2 3 4 5

//TH còn lại: (là nút cuối)

//Tìm nút cuối

Last=F;

while(Last->Link!=NULL) Last=Last->Link;

DeleteM(F,Last);//2 3 4

//Test hàm có giá trị = x

//TH là nút bất kỳ

DeleteX(F,3);// 2 4

//TH là nút cuôí

DeleteX(F,4);//2

//TH ko thấy

DeleteX(F,0); //2

//TH la nut dau

DeleteX(F,2);//Khong in ra cái gì :v

Display(F);

//---END CODE TEST---

cout<<endl;

return 0;

}

//===dinh nghia ham===

//---cac ham bo sung phan tu du lieu x vao DSLKD----

void BeforeInsert(Node\* &F, Node \*M, int x)//bo sung x vao truoc M

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Link = NULL;

//2. T/h danh sach rong

if(F==NULL)

{

F = N;

return;

}

//3. T/h M la nut dau tien

if(M==F)

{

N->Link = F;

F = N;

return;

}

//4. T/h con lai

//Cho P tro toi truoc M

Node \*P = F;

while(P->Link != M) P=P->Link;

//Thay doi lien ket

P->Link = N;

N->Link = M;

}

//-----------------------------

void PostInsert(Node\* &F, Node \*M, int x)// bo sung x vao sau nut M

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Link = NULL;

//2. T/h danh sach rong

if(F==NULL)

{

F = N;

return;

}

//3. T/h con lai

N->Link = M->Link;

M->Link = N;

}

//-----------------------------

void FirstInsert(Node\* &F, int x)// bo sung x vao dau danh sach

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Link = NULL;

//2. T/h danh sach rong

if(F==NULL)

{

F = N;

return;

}

//3. Bo sung vao dau danh sach

N->Link = F;

F = N;

}

//-----------------------------

void LastInsert(Node\* &F, int x)// bo sung x vao cuoi danh sach

{

//1. Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Link = NULL;

//2. T/h danh sach rong

if(F==NULL)

{

F = N;

return;

}

//3. Bo sung vao nut cuoi

Node \*P = F;

while(P->Link != NULL) P=P->Link;

//Thay doi lien ket

P->Link = N;

}

//-----------------------------

//---cac ham loai bo mot nut khoi DSLKD----

int FirstDelete(Node\* &F) // xoa nut dau danh sach

{

//1. Kiem tra danh sach rong

if(F==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

//2. Giu lai dia chi va gia tri cua nut dau

Node \*P = F;

int Tg = F->Infor;

//3. Thay doi lien ket

F=F->Link;

//4. Huy nut dau

delete P;

//5. Tra ve gia tri cua nut dau bi loai bo

return Tg;

}

//-----------------------------

int LastDelete(Node\* &F)//xoa nut cuoi danh sach

{

//1. Kiem tra danh sach rong

if(F==NULL)

{

cout<<"Danh sach da rong!";

return -1;

}

//2. Luu lai gia tri nut cuoi

Node \*P = F, \*Q;

//Cho P tro toi nut cuoi

while(P->Link!=NULL)

{

Q=P;//giu lai nut truoc P de tro toi nut cuoi sau khi xoa P

P=P->Link;

}

int Tg = P->Infor;

//3. Xoa nut cuoi

delete P;

//4. Cho nut cuoi tro toi NULL

Q->Link = NULL; // Fix bug stackoverflow (nút cuối trỏ linh tinh)

//5. Tra ve gia tri nut cuoi bi loai bo

return Tg;

}

//-----------------------------

int DeleteM(Node\* &F, Node \*M)// xoa nut M

{

//1. Kiem tra hang doi rong

if(F==NULL)

{

cout<<"Hang doi da rong!";

return -1;

}

//2. Giu lai gia tri cua nut M

int Tg = M->Infor;

//3. Thay doi lien ket

if(M==F) //TH M la nut dau

F = F->Link;

else //TH con lai

{

//Tim den nut truoc nut M

Node \*P = F;

while(P->Link != M) P = P->Link;

//Noi nut truoc M voi nut sau M

P->Link = M->Link;

}

//4. Huy nut M

delete M;

//5. Tra ve gia tri cua nut M bi loai bo

return Tg;

}

//-----------------------------

int DeleteX(Node\* &F, int x)// xoa nut co gia tri = x

{

//1. Kiem tra hang doi rong

if(F==NULL)

{

cout<<"Hang doi da rong!";

return -1;

}

Node \*P=F;

while(P)

{

if(P->Infor==x)

{

//TH la nut dau

if(P==F) F = F->Link;

else //TH con lai

{

//Cho Q tro toi nut truoc nut P

Node \*Q=F;

while(Q->Link!=P) Q=Q->Link;

//Noi Q voi nut sau P

Q->Link = P->Link;//Bao gom ca truong hop P la nut cuoi thi Q se la nút cuối luôn

}

//Xoa nut P co gia tri = x

delete P;

//Xoa thanh cong

return 1;

}

//Cho P tro toi nut tiep theo

P=P->Link;

}

//Khong tim thay nut nao co gia tri = x

return 0;

}

//-----------------------------

//---ham duyet danh sach---

void Display(Node\* F)

{

while(F)

{

cout<<F->Infor<<" ";

F=F->Link;

}

}

**//De: Ngan xep 10->2**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#define N 50

using namespace std;

struct Stack

{

int T;

int S[N];

};

//Khai bao ham

void Push(Stack &NS,int x);

int Pop(Stack &NS);

int IsEmpty(Stack NS);

//===Chuong trinh chinh===

int main()

{

Stack NS={-1};

int n,thuong;

cout<<"Nhap vao so nguyen n = ";cin>>n;

thuong=n;

while(thuong)

{

Push(NS,thuong%2);

thuong/=2;

}

cout<<"So nhi phan cua "<<n<<" la: ";

while(!IsEmpty(NS)) cout<<Pop(NS);

cout<<endl;

return 0;

}

//===Dinh nghia ham===

void Push(Stack &NS,int x)

{

//Kiem tra ngan xep day

if(NS.T==N-1)

{

cout<<"Ngan xep day!";

return;

}

//Thay doi chi so T va dua vao ngan xep

NS.S[++NS.T]=x;

}

int Pop(Stack &NS)

{

//Kiem tra ngan xep rong

if(NS.T==-1)

{

cout<<"Ngan xep rong!";

return -1;

}

return NS.S[NS.T--];

}

int IsEmpty(Stack NS)

{

if(NS.T==-1) return 1;

return 0;

}

**//De: Hang doi luu tru phan tan**

#include <iostream>

#include <fstream>

using namespace std;

//Khai bao bao cau truc Node

struct Node

{

int Infor;

Node \*Link;

};

//Khai bao cau truc hang doi

struct Queue

{

Node \*F, \*R;

};

//Khai bao cac ham

void QInsert(Queue &FR, int x); //Bo sung phan tu x

int QDelete(Queue &FR); //Loai bo phan tu

bool QEmpty(Queue FR); //Kiem tra rong

int Front(Queue FR); //Tra ve phan tu dau hang doi

int Rear(Queue &FR); //Tra ve phan tu cuoi hang doi

//===chuong trinh chinh===

int main()

{

//Cài đặt cấu trúc lưu trữ hàng đợi

Queue FR = {NULL, NULL};

//TEST

QInsert(FR, 1);//1

QInsert(FR, 2);//1 2

QInsert(FR, 3);//1 2 3

cout<<Front(FR)<<endl;//1

cout<<Rear(FR)<<endl;//3

while(!QEmpty(FR)) cout<<QDelete(FR)<<" ";

//END TEST

cout<<endl;

return 0;

}

//===dinh nghia ham===

void QInsert(Queue &FR, int x) //Bo sung phan tu x

{

//1.Tao nut moi

Node \*N = new Node;

N->Infor = x;

N->Link = NULL;

//2-3. Noi nut moi vao sau R

//va cho R tro toi nut moi

if(FR.R==NULL)//TH hang doi rong

FR.F = FR.R = N;

else// TH con lai

{

FR.R->Link = N;

FR.R = N;

}

}

//-------------------------------

int QDelete(Queue &FR) //Loai bo phan tu

{

//1. Kiem tra hang doi rong

if(FR.F==NULL)

{

cout<<"Hang doi da rong!";

return -1;

}

//2. Giu lai giu lieu va dia chi nut F

int Tg = FR.F->Infor;

Node \*P = FR.F;

//3. Thay doi F

if(FR.F==FR.R) //TH hang doi chi co 1 phan tu

FR.F = FR.R = NULL;

else //TH con lai

FR.F = FR.F->Link;

//4. Huy nut va tra ve gia tri

delete P;

return Tg;

}

//-------------------------------

bool QEmpty(Queue FR) //Kiem tra rong

{

if(FR.F==NULL) return 1;

return 0;

}

//-------------------------------

int Front(Queue FR) //Tra ve phan tu dau hang doi

{

//1. Kiem tra hang doi rong

if(FR.F==NULL)

{

cout<<"Hang doi da rong!";

return -1;

}

//2. Tra ve phan tu dau hang doi

return FR.F->Infor;

}

//-------------------------------

int Rear(Queue &FR) //Tra ve phan tu cuoi hang doi

{

//1. Kiem tra hang doi rong

if(FR.F==NULL)

{

cout<<"Hang doi da rong!";

return -1;

}

//2. Tra ve phan tu cuoi hang doi

return FR.R->Infor;

}

**//De: Cai dat cay nhi phan voi phan tu la ky tu**

#include<iostream>

#include<stdio.h>

using namespace std;

//Khai bao cau truc nut cua cay nhi phan

struct Node

{

char Infor;

Node \*Left,\*Right;

};

//khai bao ham

void PreOrder(Node \*T);

void InOrder(Node \*T);

void PostOrder(Node \*T);

//===chuong trinh chinh===

int main()

{

//Khai bao bien cua cau truc cay nhi phan

Node \*T;

//Tao cay

//Tao nut 1

T = new Node;

T->Infor='/';

//Tao nut 2

T->Left=new Node;

T->Left->Infor='+';

//Tao nut 3

T->Right=new Node;

T->Right->Infor = 'c';

T->Right->Left = T->Right->Right = NULL;

//Tao nut 4

T->Left->Left = new Node;

T->Left->Left->Infor = 'a';

T->Left->Left->Left = T->Left->Left->Right = NULL;

//Tao nut 5

T->Left->Right = new Node;

T->Left->Right->Infor = 'b';

T->Left->Right->Left = T->Left->Right->Right = NULL;

cout<<"\nDang tien to cua bieu thuc (a+b)/c la:\n";

PreOrder(T);

cout<<"\nDang trung to cua bieu thuc (a+b)/c la:\n";

InOrder(T);

cout<<"\nDang hau to cua bieu thuc (a+b)/c la:\n";

PostOrder(T);

cout<<endl;

return 0;

}

//===dinh nghia ham===

void PreOrder(Node \*T)

{

if(T==NULL) return;

else

{

cout<<T->Infor<<" ";

PreOrder(T->Left);

PreOrder(T->Right);

}

}

//---------------------

void InOrder(Node \*T)

{

if(T==NULL) return;

else

{

PreOrder(T->Left);

cout<<T->Infor<<" ";

PreOrder(T->Right);

}

}

//---------------------

void PostOrder(Node \*T)

{

if(T==NULL) return;

else

{

PreOrder(T->Left);

PreOrder(T->Right);

cout<<T->Infor<<" ";

}

}