#include <iostream>

#include <cstring>

#include <typeinfo>

using namespace std;

//双链表节点结构体定义

struct DuallistNode {

void \* datap;

struct DuallistNode \* prev; //指向上一个节点的指针

struct DuallistNode \* next; //指向下一个节点的指针

};

//双链表结构体定义，用来保存链表头节点指针和链表尾节点指针

struct Duallist {

struct DuallistNode \*head; //指向链表第一个节点

struct DuallistNode \*tail; //指向链表最后一个节点

};

//b 整型比较

int cmp\_int(const int\* aInt, const int\*bInt)

{

if (\*aInt>\*bInt) return 1;

if (\*aInt<\*bInt) return -1;

if (\*aInt==\*bInt) return 0;

}

//c 字符串比较

int cmp\_str(const char\* aStr, const char\* bStr)

{

return strcmp(aStr,bStr);

}

//d 删除节点

void\* delete\_node(struct Duallist \*aDuallist,

struct DuallistNode \*p)

{

void \*data;

p->prev->next=p->next,p->next->prev=p->prev;

data=p->datap;

delete p;

return data;

}

//将p的前一个节点和后一个链接并删除p就可以了吧，

//为什么还要输入一个双链表Duallist \*aDuallist？

//a 查找数据

int cmp (const void\* x, const void\* y)

{

if (typeid(x)==typeid(int\*))

return cmp\_int((int\*)x,(int\*)y);

else return cmp\_str((char \*)x,(char\*)y);

}

struct DuallistNode\* find\_data(struct Duallist \*aDuallist,

void\* data, int(\*cmp)(const void\*, const void\*))

{

DuallistNode \*p;

bool flag=false;

p=aDuallist->head;

while (p)

{

if (p->datap==data)

{

\*((int\*)(p->datap))=(\*cmp)(p->datap,data);

break;

}

p=p->next;

}

if (flag) return p;

else

{

p->next=NULL,p->prev=NULL;

return p;

}

}

//e 找hello word

int main()

{

bool flag=false;

Duallist \*aDuallist;

//cin>>aDuallist 输入数组

char pettern[]="hello world!";

DuallistNode \*p;

p=aDuallist->head;

while (p)

{

if ((char\*)(p->datap)==pettern)

flag=true;

else p=p->next;

}

if (flag) cout<<"Found";

else cout<<"Not Found";

}