Đề 1

Câu 1:

Void bublesort(int a[],int n)

{

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

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

If(a[j]>a[j+1]

{

Int temp=a[j];

A[j]=a[j+1];

A[j+1]=temp;

}

}

Độ phức tạp: O(n^2)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Ban đầu | 2 | 17 | 3 | 1 | 9 | 3 | 2 |
| Lần 1 | 2 | 3 | 1 | 9 | 3 | 2 | 17 |
| Lần 2 | 2 | 1 | 3 | 3 | 2 | 9 | 17 |
| Lần 3 | 1 | 2 | 3 | 2 | 3 | 9 | 17 |
| Lần 4 | 1 | 2 | 2 | 3 | 3 | 9 | 17 |

Câu 2

#include<stdio.h>

Typedef struct

{

Char ten[50];

Int ma;

Float hsl;

Int namsinh;

}nv;

Typedef struct NODE

{

NODE\*link;

Nv data;

}node;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

Void sapxep(llist\*l)

{

Node\*i,node\*j;

For(i=l->phead;i!=NULL;i=i->link)

For(j=i->link;j!=NULL;j=j->link)

{

If(i->data.namsinh<j->data.namsinh)

{

Nv temp=i->data;

i->data=j->data;

j->data=temp;

}

Else

If(i->data.namsinh==j->data.namsinh

{

If(i->data.ma<j->data.ma)

{

Nv temp=i->data;

i->data=j->data;

j->data=temp;

}

}

}

}

Câu 3:

Int n=783;

Int s=0;

While(n!=0)

{

S+=N%10;

N/=10;

}

Int i;

For(i=0;i<=s/2;i++)

{

If(s%i==0)

Printf(“S không là số nguyên tố’’);

Return 0;

}

Printf(“S là số nguyên tố”);

Return 0 ;

Câu 4 :

a)

Typedef struct

{

Char ten[50];

Int socmt;

Char diachi[100];

}thongtin;

Typedef struct NODE

{

NODE\*left ;

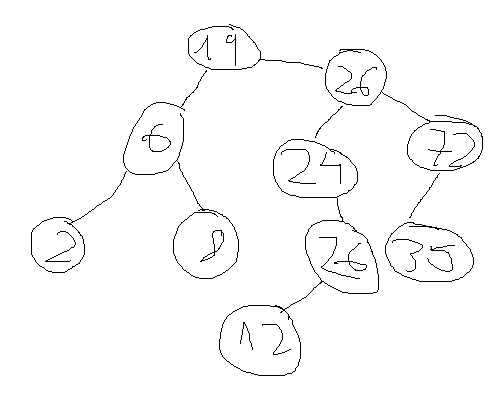
NODE\*right ;

thongtin data ;

}node ;

b)

Cách chèn : cây rỗng thì giá trị chuyền vào là gốc luôn, giá trị lớn sang bên phải, nhỏ sang trái



Void LNR( llist l)

{

LNR(l->left) ;

Printf(‘‘%d ’’,l->data);

LNR(l->right);

}

In cây: 2,6,8,19,24,12,26,26,35,72

Đề 2

Câu 1:

Void doicho(int a[],int n)

{

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

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

If(a[i]>a[j]

{

Int temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

Độ phức tạp: O(n^2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Ban đầu | 16 | 4 | 8 | 19 | 30 | 26 | 7 | 1 |
| Lần 1 | 1 | 16 | 8 | 19 | 30 | 26 | 7 | 4 |
| Lần 2 | 1 | 4 | 16 | 19 | 30 | 26 | 8 | 7 |
| Lần 3 | 1 | 4 | 7 | 19 | 30 | 26 | 16 | 8 |
| Lần 4 | 1 | 4 | 7 | 8 | 30 | 26 | 19 | 16 |
| Lần 5 | 1 | 4 | 7 | 8 | 16 | 30 | 26 | 19 |
| Lần 6 | 1 | 4 | 7 | 8 | 16 | 19 | 30 | 26 |
| Lần 7 | 1 | 4 | 7 | 8 | 16 | 19 | 26 | 30 |

Câu 2:

Typedef struct

{

Int msv;

Char ten[50];

Float dtb;

}sv;

Typedef struct NODE

{

NODE\*link;

Sv data;

}node;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

Void xoa(llist\*l,int x)

{

If(l->spt==0)

Return ;

Node\*i ;

For(i=l->phead ;i!=NULL;i=i->link)

{

If(i->

}

}

Câu 3:

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

{

If(a[i]%2!=0)

{

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

{

If(a[i]>a[j]&&a[j]%2!=0

{

Int temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

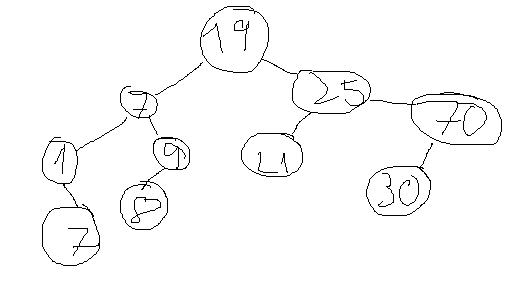
}

}

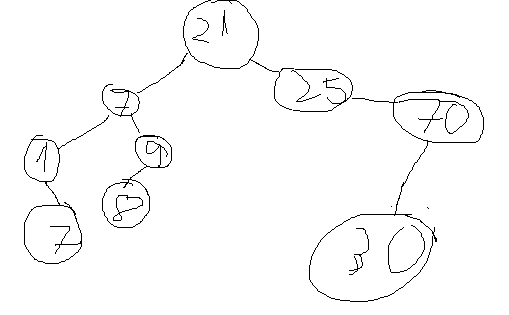
Kết quả: {-9,6,-1,4,3,5}

Câu 4:

Cách chèn : cây rỗng thì giá trị chuyền vào là gốc luôn, giá trị lớn sang bên phải, nhỏ sang trái



Xóa gốc:



Void insert(tree\*t, sv x)

{

If(t==NULL)

{

Node\*p=(node\*)malloc(sizeof(node));

p->data=x;

p->left=NULL;

p->right=NULL;

}

Else

{

If(x>t->data)

{

Insert(t->right,x);

}

Else

If(x<t->data)

{

Insert(t->left,x);

}

}

}

Đề 3

Câu 1:

Quick sort bỏ

Độ phức tạp: O(nlog(n))

Câu 2:

Typedef struct

{

Char ten[50];

Int ma;

Bool gt;

}sv;

Typedef struct NODE

{

NODE\*link;

Sv data;

}node;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

Void inserttail (llist\*l, sv x)

{

Node\*p=(node\*)malloc(sizeof(node));

p->data=x;

p->link=NULL;

If(l->spt==0)

l->phead=l->ptail=p;

else

{

l->ptail->link=p;

l->ptail=p;

}

l->spt++;

}

Void tach(llist l)

{

Llist l1;// danh sách nam

Llist l2;// danh sách nữ

Node\*I;

For(i=l.phead;i!=NULL;i=i->link)

{

If(i->data.gt==1)// 1 là nam

Inserttail(l1,i->data);

Else

If(i->data.gt==0)// 0 là nữ

Inserttail(l2,i->data);

}

}

Câu 3:

Tự chuyển

Câu 4:

Typedef struct

{

Int ma;

Char ten[50];

Float luong;

}nv;

Typedef struct NODE

{

NODE\*left;

NODE\*right;

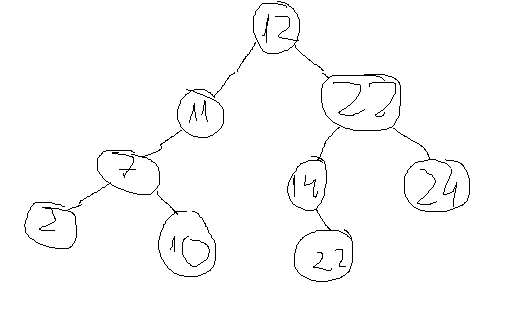
Nv data;

}node;

Typedef struct node\*tree;

b)

Cách chèn : cây rỗng thì giá trị chuyền vào là gốc luôn, giá trị lớn sang bên phải, nhỏ sang trái



Đề 4

Câu 1:

Void chen(int a[],int n)

{

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

{

Int temp=a[i];

Int j

For(j=i-1;j>=0;j--)

{

If(a[j]<temp)

{

a[j+1] = a[j];

}

Else

Break;

}

a[j+1] = temp;

}

}

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Ban đầu | 19 | 7 | 16 | 2 | 6 | 4 | 20 | 18 | 12 |
| Lần 1 | 7 | 19 | 16 | 2 | 6 | 4 | 20 | 18 | 12 |
| Lần 2 | 7 | 16 | 19 | 2 | 6 | 4 | 20 | 18 | 12 |
| Lần 3 | 2 | 7 | 16 | 19 | 6 | 4 | 20 | 18 | 12 |
| Lần 4 | 2 | 6 | 7 | 16 | 19 | 4 | 20 | 18 | 12 |
| Lần 5 | 2 | 4 | 6 | 7 | 16 | 19 | 20 | 18 | 12 |
| Lần 6 | 2 | 4 | 6 | 7 | 16 | 19 | 20 | 18 | 12 |
| Lần 7 | 2 | 4 | 6 | 7 | 16 | 18 | 19 | 20 | 12 |
| Lần 8 | 2 | 4 | 6 | 7 | 12 | 16 | 18 | 19 | 20 |

Bài 2:

Typedef struct

{

Char ten[50];

Int gia;

Int sl;

}mh;

Typedef struct NODE

{

NODE\*link;

Mh data;

}node;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

Void sapxep(llist \*l)

{

Node\*i,\*j;

For(i=l->phead;i!=NULL;i=i->link)

For(j=i->link;j!=NULL;j=j->link)

{

If(i->data.gia>j->data.gia)

{

Mh temp=i->data;

i->data=j->data;

j->data=temp;

}

Else

If(i->data.gia==j->data.gia)

{

If(strcmp(i->data.ten,j->data.ten)>0)

{

Mh temp=i->data;

i->data=j->data;

j->data=temp;

}

}

}

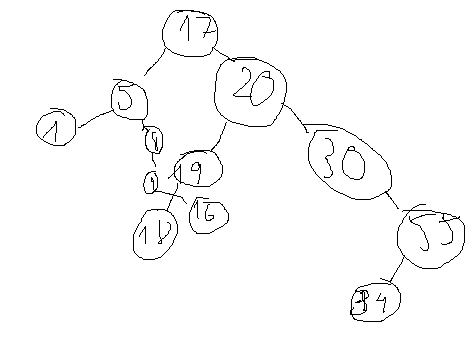
}

Câu 3:

Sơ đồ tự vẽ

Kết quả: {5,6,-1,4,-9,3}

Câu 4:



Cách xóa 1 nút khỏi cây:

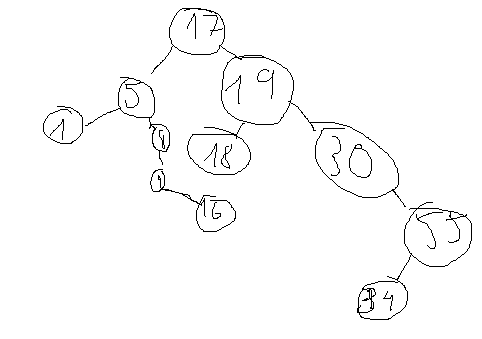
+ Duyệt xem có nút cần tìm hay ko, nếu ko thì thoát luôn.

+ Nếu nút cần xóa không có nút con thì xóa trực tiếp luôn.

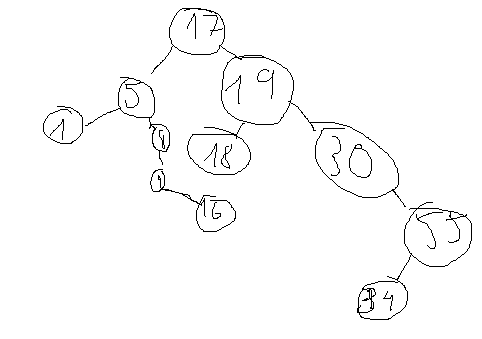
+ Nếu nút cần xóa có 1 con thì xóa liên kết giữa nút cần xóa và nút con sau đó liên kết nút con với nút cha của nút cần xóa và xóa nút cần xóa

+Nếu nút cần xóa có 2 con thì tìm con phải đầu tiên của cây con trái và thế chỗ do nút cần xóa.Nếu cây con trái không có nút phải thì đẩy nút con trái lên thế chỗ nút cần xóa.

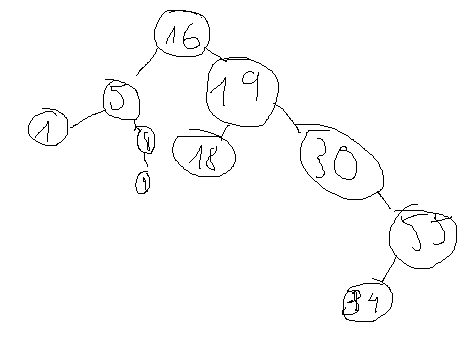
Cây sau khi xóa khóa 20



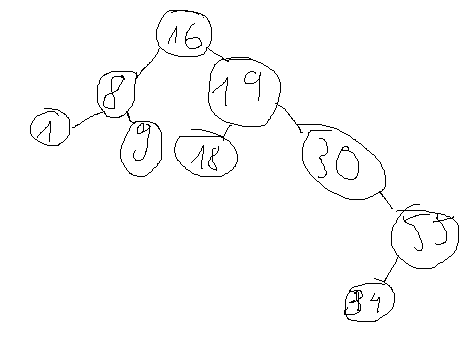
Cây sau khi xóa khóa 11



Cây sau khi xóa khóa 17



Cây sau khi xóa khóa 5



Trung tố: 12\*5 + 2\*(17-3\*4) + 8

Tiền tố: ++\*12 5 2\*-17 \*3 4 8

Hậu tố: 12 5 \* 2 17 3 4 \*-\*+8+

Đề 5

Câu 1:

Int timkiem (int a[],int l,int r,int x)

{

Int m=(l+r)/2;

If(a[m]==x)

Return m;

Else

If(l>m||r<m)

Return -1;

Else

If(a[m]>x)

Return Timkiem(a,l,m-1,x);

Else

If(a[m]<x)

Return Timkiem(a,m+1,r,x);

}

Câu 2:

Typedef struct

{

Char ten[50];

Int thamnien;

Int hsl;

Int luong;

}ns;

Typedef struct NODE

{

NODE\*next;

NODE\*prev;

Ns data;

}note;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

Void inserttail(llist\*l,ns a)

{

Node\*p=(node\*)malloc(sizeof(node));

p->data=a;

p->next=NULL;

p->prev=NULL;

If(l->spt==0)

l->phead=l->ptaid=p;

else

{

l->ptail->Link=p;

p->prev=l->ptail;

l->ptail=p;

}

l->spt++;

}

Câu 3:

Int d=0,i,n=6;

For(i=n-1;i>=0;i--)

{

If(a[i]%2==0)

{

Printf(“%d, %d”, a[i], i);

d=1;

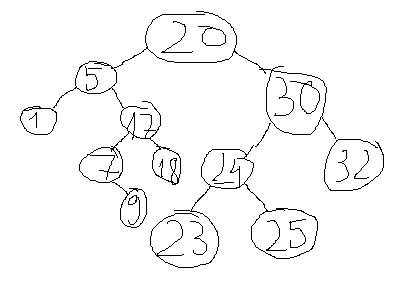
}

}

If(d==0)

Printf(“Không có số thỏa mãn”);

Câu 4:



Cách thức xóa 1 nút khỏi cây:

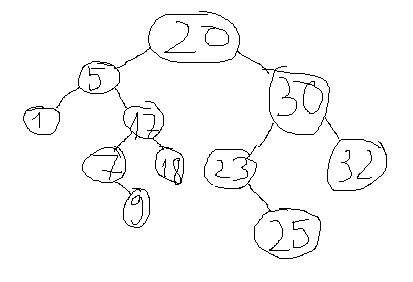
+ Duyệt xem có nút cần tìm hay ko, nếu ko thì thoát luôn.

+ Nếu nút cần xóa không có nút con thì xóa trực tiếp luôn.

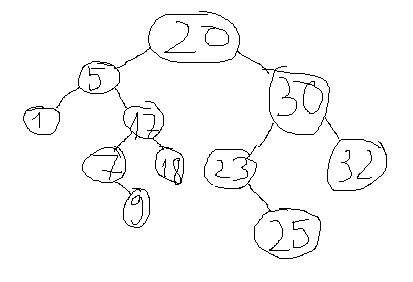
+ Nếu nút cần xóa có 1 con thì xóa liên kết giữa nút cần xóa và nút con sau đó liên kết nút con với nút cha của nút cần xóa và xóa nút cần xóa

+Nếu nút cần xóa có 2 con thì tìm con phải đầu tiên của cây con trái và thế chỗ do nút cần xóa.Nếu cây con trái không có nút phải thì đẩy nút con trái lên thế chỗ nút cần xóa.

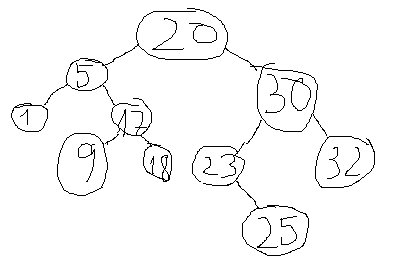
Cây sau khi xóa nút chứa khóa 24:



Cây sau khi xóa nút có khóa là 10



Cây sau khi xóa nút có khóa là 7



Đề 6

Câu 1:

Void bublesort(sv a[])

{

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

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

{

If(a[j].tuoi>a[j+1].tuoi)

Sv temp=a[j];

a[j]=a[i];

a[j]=temp;

else

if(a[j].tuoi==a[j+1].tuoi)

{

If(strcmp(a[j].tuoi,a[j+1].tuoi)>0)

{

Sv temp=a[j];

a[j]=a[i];

a[j]=temp;

}

}

}

}

Câu 2:

Typedef struct

{

Char ten[50];

Int mns;

Float hsl;

Int luongcb;

}ns;

Typedef struct NODE

{

NODE\*left;

NODE\*right;

Ns data;

}node;

Typdef struct node\*tree;

Void insert(tree\*t, ns a)

{

Node\*p=(node\*)malloc(sizeof(node));

p->left=NULL;

p->right=NULL;

p->data=a;

If(t==NULL)

T=p;

Else

{

…..

}

}

Câu 3:

Int k=3;int I;int b=100;int n=6;

For(i=n-1;i>=k+1;i--)

{

A[i]=a[i-1];

}

A[k]=b;

For(i=0;i<n;i++)

{

Printf(“%d ”,a[i];

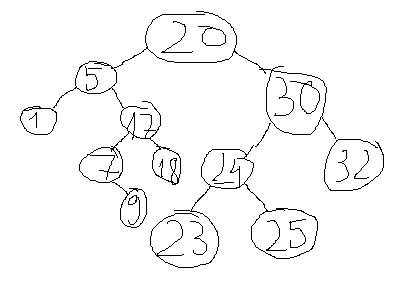
}

Kết quả sau khi thực hiện:

{5,6,-1,100,4,-9}

Câu 4:

Cách chèn : cây rỗng thì giá trị chuyền vào là gốc luôn, giá trị lớn sang bên phải, nhỏ sang trái



Những số như 1 với 31 không chèn vì đã có trong cây rồi

Cây sau khi xóa:

>>>

B)

Trung tố : x\*y + z\*(t-u)

Tiền tố : + \* x y \* z - t u

Hậu tố : x y \* z t u - \* +

Đề 7

Câu 1 :

int timkiem(int a[],int l, int r,int x)

{

Int m=(l+r)/2;

If(a[m]==x)

Return m;

Else

If(l>m||r<m)

Return -1;

Else

If(a[m]>x)

Return timkiem(a,l,m-1,x);

Else

If(a[m]<x)

Return timkiem(a,m+1,r,x);

}

Tìm kiếm như nào thì xem video

Câu 2:

Typedef struct

{

Char ten[50];

Int mns;

Float hsl;

Int luongcb;

}ns;

Typedef struct NODE

{

NODE\*left;

NODE\*right;

Ns data;

}node;

Typdef struct node\*tree;

Float tinhluong(ns a)

{

Return lcb\*hsl;

}

Câu 3:

Tự xem video

Câu 4:

Hậu tố: 12 30 48 2 / 29 - \* + 37 17 - 2 / +

Trung tố: (12+ 30\* (48/2-29)) + (37-17)/2= -128

Tiền tố: + + 12 \* 30- / 48 2 29 / – 37 17 2

Đề 8

Câu 1:

Void sapxep(int a[],int n)

{

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

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

If(a[i]>a[j]

{

Int temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

Thực hiện : xem video

Câu 2:

Typedef struct

{

Char ten[50];

Int ma;

Float diemtb;

}sv;

Typedef struct NODE

{

NODE\*link;

Sv data;

}node;

Typedef struct LLIST

{

Node\*phead;

Node\*ptail;

Int spt;

}llist;

llist\* GopDS(llist L1,llist L2)

{

if(L1.spt==0 && L2.spt==0)

return &L1;

if(L1.spt==0)

return &L2;

if(L2.spt==0)

return &L1;

L1.spt+=L2.spt;

L1.pTail->link=L2.pHead;

L1.pTail=L2.pTail;

return &L1;

}

Câu 3:

Xem video+xem ảnh

Câu 4 :

Tiền tố: + + 12 \* 30 - / 48 2 29 / - 37 17

Trung tố: (12+30\* (48/2-29))+ (-37/17)

Hậu tố: 12 30 48 2 / 29 - \* + -37 17 / +