include irvine32.inc

.data

insertnodeBST proto , a:dword

delete proto ,x:dword,y:dword,z:dword

Searching proto , q:dword , e:dword , s:dword

PreOrderTraversal proto ,e:dword

visit proto,l:dword

;ltr1 byte "Node successfully created! ",0

ltr2 byte "No node available.",0

ltr3 byte "Tree Deleted!!",0

ltr4 byte "In Order Triversal OF Binary Tree Is:.",0

ltr5 byte "Number Found! ",0

ltr6 byte "Not Found! ",0

ltr7 byte "Search Number : ",0

ltr9 byte "Deleting :",0

ltr8 byte "::::::::::::::::::BINARY SEARCH TREE :::::::::::::::::::.",0

vsearch dword 0

\_size = 100

NULL = 0

count dword 0

root dword 0

n dword 0

i dword 0

lchild dword 0

rchild dword 0

currnode dword ?

handler dword ?

tempoo dword ?

value dword ?

dwFlags DWORD HEAP\_ZERO\_MEMORY

.code

main proc

invoke getprocessheap

mov handler , eax

mov ebx,0

mov edx,offset ltr8

call writestring

call crlf

invoke insertnodeBST , 34

invoke insertnodeBST , 22

invoke insertnodeBST , 42

invoke insertnodeBST , 10

invoke insertnodeBST , 23

invoke insertnodeBST , 4

call print

mov vsearch ,55

invoke Searching , root , count , vsearch

;call print

mov value,10

invoke delete , root , count , value

call print

push root

push count

call InOrderTraversal

call print

call crlf

main endp

insertnodeBST proc , a: dword

cmp root ,0

je j1

mov ebx,a

cmp root , ebx

jg j2

jmp j3

j2:

mov ebx,a

mov lchild,ebx

mov [esi],ebx

jmp quitbst

j3:

mov ebx,a

mov rchild,ebx

mov [esi],ebx

jmp quitbst

j1:

invoke heapalloc , handler , dwFlags , \_size

mov root , eax

mov currnode ,eax

mov esi,currnode

mov eax,a

mov [esi],eax

quitbst:

; mov edx,OFFSET ltr1

;call writestring

;call crlf

add esi,4

inc count

ret

insertnodeBST endp

print proc

mov ecx , count

mov esi, root

cmp count,0

je j5

j4:

mov eax ,[esi]

call writedec

call crlf

add esi,4

loop j4

jmp j6

j5:

mov edx,offset ltr2

call writestring

call crlf

j6:

ret

print endp

InOrderTraversal proc

push ebx

mov ebx,esp

mov ecx,[ebx+8]

mov edx,offset ltr4

call writestring

call crlf

dec ecx

l1:

push ecx

mov eax,[ebx+12]

;mov lchilld,[ebx+12]

mov esi,eax

l2:

mov eax,[esi]

mov rchild,eax

cmp eax,[esi+4]

jb l3

xchg eax,[esi+4]

mov [esi],eax

l3:

add esi,4

loop l2

pop ecx

loop l1

pop ebp

ret 8

InOrderTraversal endp

DeleteTree Proc

INVOKE HeapFree, handler, dwFlags, root

mov edx,OFFSET ltr3

call writestring

call crlf

mov count,0

mov root,NULL

ret

DeleteTree endp

Delete proc ,t:dword,k:dword ,v:dword

mov edx,offset ltr9

call writestring

mov eax,v

call writedec

call crlf

mov esi,root

mov ecx,k

del:

mov ebx,[esi]

mov eax,v

cmp eax,ebx

je p2

add esi,4

loop del

jmp d

p2:

mov edx,[esi+4]

mov ebx,0

cmp edx,ebx

je p3

cmp [esi+8],ebx

jne bold

mov [esi],edx

jmp d

p3:

mov edx,[esi+8]

mov ebx,0

cmp edx,ebx

je d

bold:

mov ecx,[esi+12]

mov ebx,0

cmp ecx,ebx

jne m

mov ebx,[esi+8]

mov [esi],ebx

d:

dec count

ret

m:

mov ebx,[esi+8]

mov [esi],ebx

ret

delete endp

Searching proc , q:dword , e:dword , s:dword

mov esi,q

mov ecx,e

mov ebx,s

mov edx,offset ltr7

call writestring

mov eax,s

call writedec

call crlf

op:

mov eax,[esi]

cmp eax,ebx

je found

add esi,4

loop op

mov edx,offset ltr6

call writestring

call crlf

ret

found:

mov edx,offset ltr5

call writestring

call crlf

ret

searching endp

visit proc , f:dword

mov eax,f

call writedec

call crlf

ret

visit endp

end main