include irvine32.inc

.data

str1 byte "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DOUBLY LINKED LIST\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",0

str2 byte " Obaid Ur Rehman",0

str3 byte " Hina Waheed",0

str4 byte " Moosa Hussain",0

str5 byte "Head Inserted! ",0

str6 byte "No Nodes Found! ",0

str7 byte "All nodes deleted.",0

str8 byte "Last node inserted",0

str9 byte "Last node deleted",0

str10 byte "First node deleted.",0

str11 byte "Node inserted at position: ",0

str12 byte "Node deleted at position: ",0

insertatheadDLL proto , a:dword

insertlastDLL proto ,a:dword

deletenodesDLL proto

insertFromLastDLL proto , a:dword

removelastDLL proto

removefrontDLL proto

removefromposDLL proto,pos:dword

insertatposDLL proto,key:dword,pos:dword

head dword 0

tail dword 0

nextptr dword 0

\_size = 100 ; 25 nodes of dword

NULL = 0

count dword 0

handler dword ?

dwFlags DWORD HEAP\_ZERO\_MEMORY

.code

main PROC

mov edx,OFFSET str1

call writestring

call crlf

mov edx,OFFSET str2

call writestring

call crlf

mov edx,OFFSET str3

call writestring

call crlf

mov edx,OFFSET str4

call writestring

call crlf

invoke getprocessheap

mov handler , eax

call printDLL

invoke insertatheadDLL , 5

invoke insertatheadDLL , 8

invoke insertatheadDLL , 48

invoke insertatheadDLL , 52

invoke insertatheadDLL,89

invoke insertatposDLL,3,578

invoke removefromposDLL,2

call printDLL

exit

main ENDP

insertatheadDLL proc , a:dword

cmp head , NULL

je l1

mov ecx, count

mov esi,head

mov ebx,[esi+4]

mov eax,[esi]

mov [esi+4] , eax

l:

add esi,4

mov tail,esi

mov eax,ebx

mov ebx,[esi+4]

mov [esi+4],eax

loop l

mov esi,head

mov eax,a

mov [esi],eax

jmp quit

l1:

invoke heapalloc , handler , dwFlags , \_size

mov head,eax

mov esi,head

mov tail,esi

mov eax,a

mov [esi],eax

jmp quit

quit:

inc count

mov edx,OFFSET str5

call writestring

call crlf

ret

insertatheadDLL endp

printDLL proc

mov ecx,count

mov esi,head

cmp head , NULL

je l2

l1:

mov eax,[esi]

call writedec

call crlf

add esi , 4

;mov nextptr , eax

loop l1

jmp quit

l2:

mov edx,OFFSET str6

call writestring

call crlf

quit:

ret

printDLL endp

deletenodesDLL proc

INVOKE HeapFree, handler, dwFlags, head

mov edx,OFFSET str7

call writestring

call crlf

mov count,0

mov head,NULL

mov tail,NULL

ret

deletenodesDLL endp

insertlastDLL proc a:dword

cmp head , NULL ; if head is null

je l1

mov esi,tail

mov eax,a

add esi,4

mov [esi],eax

mov tail,esi

jmp quit

l1:

invoke heapalloc , handler , dwFlags , \_size

mov head,eax

mov esi,head

mov tail,esi

mov eax,a

mov [esi],eax

jmp quit

quit:

call crlf

mov edx,OFFSET str8

call writestring

call crlf

inc count

ret

insertlastDLL endp

removelastDLL proc

cmp head,NULL

je l1

cmp count,1

je l2

dec count

mov esi,tail

sub esi,4

mov tail,esi

jmp l3

l1:

mov edx,offset str6

call writestring

call crlf

jmp l3

l2:

call deletenodesDLL

l3:

mov edx,offset str9

call writestring

call crlf

ret 8

removelastDLL endp

removefrontDLL proc

cmp head,NULL

je l1

cmp count,1

je l2

mov esi,head

add esi,4

mov head,esi

dec count

jmp l3

l1:

mov edx,offset str6

call writestring

call crlf

jmp l3

l2:

call deletenodesDLL

l3:

mov edx,offset str10

call writestring

call crlf

ret

removefrontDLL endp

insertatposDLL proc , key: dword , pos:dword

local temp:dword

mov eax,4

mov ebx,key

mul ebx

mov esi,head

add esi,eax

mov temp,esi

mov eax,[esi]

cmp eax , 0

je l1

cmp eax,count

je l2

mov ecx, count

mov ebx,[esi+4]

mov eax,[esi]

mov [esi+4] , eax

l:

add esi,4

mov eax,ebx

mov ebx,[esi+4]

mov [esi+4],eax

loop l

mov eax,pos

mov esi,temp

mov [esi],eax

jmp quit

l1:

invoke insertatheadDLL,key

jmp quit

l2:

invoke insertlastDLL,key

quit:

mov edx,OFFSET str11

call writestring

mov eax,key

call writedec

call crlf

inc count

ret 4

insertatposDLL endp

removefromposDLL proc , pos:dword

cmp head,NULL

je nonode

mov eax,4

mov ebx,pos

mul ebx

mov esi,head

add esi,eax

mov eax,[esi]

cmp eax , 0 ; if that index is null

je l1

cmp eax,count

je l2

mov ecx,count

l3:

mov eax,[esi+4]

mov [esi],eax

add esi,4

loop l3

mov edx,OFFSET str12

call writestring

mov eax,pos

call writedec

call crlf

dec count

jmp quit

l1:

call removefrontDLL

jmp quit

l2:

call removelastDLL

jmp quit

nonode:

mov edx,OFFSET str6

call writestring

call crlf

quit:

ret

removefromposDLL endp

end main