Лабораторная работа №14

***Методы сортировки, вставки и удаления элементов таблиц.***

Выполнил ст. гр.

ІО-83, Гулевский Денис

## Варіанти виконання роботи: 306

1. Вывести на экран фамилию, группу и № варианта.

2. Определить таблицу с двумя полями:

- ключ (тип определяется по варианту)

- характеристика (тип произвольный)

Количество записей в таблице определяется по варианту.

3. Отсортировать таблицу заданным по варианту методом сортировки.

4. Произвести вставку записи в таблицу по варианту со значениями полей:

- ключ = № варианта

- характеристика = № группы

5. Произвести удаление записи из таблицы по порядковому номеру (индексу), определяемому по варианту.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 306 | *Метод сортировки*: Шейкерная | *Вставка записи*:В конец таблицы | *Кол-во записей*: 5 | *Тип ключа*: 2 символа |

.386 ;306

ASSUME CS:CODE, DS:DATA, ES:DATA, SS:MYSTACK

sELEMENT STRUC

key DW 0

field DB 0

sELEMENT ENDS

v\_table struc

v dw 0

v\_table ends

makehashkey macro value,hashkey

xor ax,ax

xor dx,dx

mov ax, value

div hash\_table\_size

mov hashkey,dx

endm

pushtotable macro key, value

local l3

makehashkey key,cx

xor si,si

lea di, hash\_table

l3:

add si,2

loop l3

add di, si

mov dx,value

mov [di].v,dx

endm

mDELETE MACRO position, tableSize, tableEnd

LOCAL L1

LEA SI, tableEnd - ((tableSize - position) + 1) \* sElemTYPE

MOV DI, SI

ADD DI, sElemTYPE

MOV CX, tableSize - position + 1

L1:

MOV AX, [DI].key

MOV BL, [DI].field

MOV [SI].key, AX

MOV [SI].field, BL

ADD DI, sElemTYPE

ADD SI, sElemTYPE

LOOP L1

tableSize = tableSize - 1

tableEnd = tableEnd - sElemTYPE

ENDM

mINSERT MACRO inKey, inField, tableSize, tableEnd

LEA SI, tableEnd

MOV AX, inKey

MOV BL, inField

MOV [SI].key, AX

MOV [SI].field, BL

tableSize = tableSize + 1

tableEnd = tableEnd + sElemTYPE

ENDM

DATA SEGMENT USE16

SZap db "Gulevsky, io-83, 306", 0Dh, 0Ah, "$"

SZap2 db "not detected", 0Dh, 0Ah, "$"

buf dw ?

hash\_table\_size dw 8

hash\_table dw '0' dup (8)

mytable v\_table <0>

v\_table <0>

v\_table <0>

v\_table <0>

v\_table <0>

v\_table <0>

v\_table <0>

v\_table <0>

sTB sELEMENT< 'ab', 1 >

sELEMENT< 'dd',2>

sELEMENT< 'cc', 3 >

sELEMENT< 'bb', 4 >

sELEMENT< 'aa', 5 >

sTBEND = $

sTBSIZE = (sTBEND - sTB)/TYPE sTB

sElemTYPE = TYPE sTB

DASH01 DW 7 DUP(0)

DATA ENDS

MYSTACK SEGMENT USE16 STACK

db 256 DUP(?)

MYSTACK ENDS

CODE SEGMENT USE16

beg:

mov ax, DATA

mov ds, ax

mov es, ax

;-----------------hash----------------------

lea si,hash\_table

mov cx,hash\_table\_size

mov di,0

mov bx,offset mytable

l1:

mov [si],bx

inc si

inc si

add bx,2

loop l1

pushtotable 'ab', 1

pushtotable 'ta', 4

pushtotable 'cc', 3

pushtotable 'dd', 2

pushtotable 'ee', 5

;--------------------------------------------

mov ah, 09h

lea dx, SZap

int 21h

;-------------SHACKER SORT------------------

LEA DI, sTB

MOV DX, sTBSIZE

CLD

CMP DX, 1

JBE bs\_exit

DEC DX

bs\_l1:

MOV CX, DX

XOR BX, BX

MOV SI, DI

bs\_l2:

LODSW

INC SI

CMP AX, [SI].key

JLE no\_XCHG1

XCHG AX, [SI].key

MOV [SI-TYPE sELEMENT].key, AX

MOV AL, [SI].field

XCHG AL, [SI-TYPE sELEMENT].field

MOV [SI].field, AL

INC BX

JMP no\_XCHG1

bs\_l3:

MOV CX, DX

XOR BX, BX

lea si, stB

add si, (sTBSIZE\*TYPE sTB)-sElemTYPE

bs\_l4:

mov ax, [SI-sElemTYPE].key

CMP AX, [SI].key

JLE no\_XCHG2

XCHG AX, [SI].key

MOV [SI-TYPE sELEMENT].key, AX

MOV AL, [SI].field

XCHG AL, [SI-TYPE sELEMENT].field

MOV [SI].field, AL

INC BX

JMP no\_XCHG2

no\_XCHG2:

sub si, 3

LOOP bs\_l4

OR BX, BX

JNZ bs\_l1

JMP bs\_exit

no\_XCHG1:

LOOP bs\_l2

OR BX, BX

JNZ bs\_l3

JMP bs\_exit

bs\_exit:

mINSERT 'ee', 5, stBSIZE, stBEND

mDELETE 1, stBSIZE, stBEND

;---------------LINEAR SEARCH-----------------

lea si, stB

mov cx, stBSIZE

mov bx, 'ab'

l:

cmp bx, [si].key

JE good

JL e

add si, 3

loop l

e:

mov ah, 09h

lea dx, SZap2

int 21h

jmp ex

good:

mov dl,[si].field

ex:

;-------------BINARY SEARCH------------------

lea di, stB

lea dx, stBEND-sElemTYPE

add si, (stBEND-stB)/2

dec si

mov bx, 'cc'

l2:

mov si, di

mov buf, dx

add buf, sElemTYPE

sub buf, di

xor eax, eax

mov ax, buf

push bx

xor bx, bx

mov bl, 2

idiv bl

xor ah,ah

pop bx

add si,ax

sub si, 1

nn:

cmp bx, [si].key

mov ax, [si].key

JE good2

JL lower

mov di,si

add di, sElemTYPE

cmp dx, di

jne next2

mov si, di

jmp nn

next2:

sub di, sElemType

cmp dx, di

je notfind

JMP l2

lower:

mov dx,si

sub dx, sElemTYPE

cmp dx, di

jne next1

mov si, di

jmp nn

next1:

add dx, sElemType

cmp dx, di

je notfind

JMP l2

notfind:

mov ah, 09h

lea dx, SZap2

int 21h

jmp ex2

good2:

mov dl, [si].field

ex2:

;-----------------HASH SEARCH----------------

xor ax,ax

xor dx,dx

makehashkey 'ee',cx

xor si,si

lea di, hash\_table

hash:

add si,2

loop hash

add di, si

mov dx,[di].v

mov AX, 4c00h

int 21h

CODE ENDS

END beg