**Лабораторна робота №10**

Виконав Бедь А.М.

**Завдання на лабораторну роботу**

У роботі реалізувати лінійний, бінарний і хеш-пошук. .

.386

makehashkey macro value,hashkey

xor ax, ax

mov al, value

div hashTableSize

mov hashkey, ah

endm

pushtotable macro value

makehashkey value,ch

xor si, si

mov dx, cx

and cx, 0ff00h

shr cx, 8

mov si, cx

shl si, 1

mov bx, hashTable[si]

mov dl, value

mov dh, 0

mov [bx], dx

endm

i equ 6

ASSUME CS:code, ds:data

mainStruct STRUC

key db 0

field db 0

mainStruct ends

data SEGMENT USE16

ts mainStruct <30h, 53h>, <31h, 56h>, <32h, 57h>, <33h, 58h>, <34h, 57h>, <35h, 57h>

endT equ $ - ts

sElement mainStruct <32h, 57h>

poz dw ? , "$"

notFindString db "not found ","$"

hashTableSize db i

hashTable dw '0' dup(i)

data ends

code SEGMENT USE16

beg:

mov ax, data

mov ds, ax

xor si, si

mov cx, endT/2

mov al, sElement.key

linearFind:

cmp ts[si].key, al

je keyEqualse

add si, 2

loop linearFind

jmp notFind

keyEqualse:

add si, 30h

mov poz, si

mov dx, offset poz

MOV ah, 09h

INT 21h

jmp endLinearFind

notFind:

mov dx, offset notFindString

MOV ah, 09h

INT 21h

endLinearFind:

xor si, si

mov cx, endT/2

mov al, sElement.key

cmp ts[si].key, al

jg notFindOrdered

cmp ts[endT - 2].key, al

jl notFindOrdered

linearFindOrdered:

cmp ts[si].key, al

je keyEqualseOrdered

add si, 2

loop linearFindOrdered

jmp notFindOrdered

keyEqualseOrdered:

add si, 30h

mov poz, si

mov dx, offset poz

MOV ah, 09h

INT 21h

jmp endLinearOrderedFind

notFindOrdered:

mov dx, offset notFindString

MOV ah, 09h

INT 21h

endLinearOrderedFind:

lea bp, ts

mov di, bp

add di, endT

mov bl, sElement.key

binarySearch:

mov ax, di

sub ax, bp

xor dx, dx

mov cx, 4h

div cx

mov cx, 2h

mul cx

mov si, bp

add si, ax

cmp bl, [si].key

je keyEqualseBinary

jl l1

jg l2

l1:

mov di, si

jmp notL2

l2:

mov bp, si

add bp, 2h

notL2:

cmp di, bp

je notFoundnBinary

jmp binarySearch

keyEqualseBinary:

add si, 30h

mov poz, si

mov dx, offset poz

MOV ah, 09h

INT 21h

jmp endBinarySearch

notFoundnBinary:

mov dx, offset notFindString

MOV ah, 09h

INT 21h

endBinarySearch:

lea si, hashTable

xor cx, cx

mov cl, hashTableSize

mov di, 0

lea bx, ts

lable:

mov [si], bx

add bx, 2

add si, 2

loop lable

xor cx, cx

int 3h

xor di, di

toHash:

pushtotable ts[di].key

add di, 02h

cmp di, endT

jge endToHash

jmp toHash

endToHash:

searchHash:

makehashkey sElement.key, cl

xor si, si

and cx, 00ffh

mov si, cx

mov ax, hashTable[si]

shr ax, 8

add ax, 30h

mov poz, ax

mov dx, offset poz

MOV ah, 09h

INT 21h

noFoundHash:

mov ax, 4c00h

INT 21h

code ends

end beg