**Лаба 1:**

Lea– получение эффективного адреса (смещения) источника

JG/JL – переход, если больше/меньше

SAR – арифметический сдвиг операнда вправо.

Все функции DOS вместо имени фуннкции имеют код, который описан в документации. Функции DOS вывзываеются через прерывание с номером 21h - int 21h. Номер функции (код (4C)) заносится в регистр ah, остальные регистры заполняются согласно спецификации на функцию (0) .

**Лаба 2:**

CLD/STD – сброс/выставление флага направления DI.

LODSB – переписывает байт, чей адрес в памяти определяется парой регистров DS:SI, в регистр al.

STOSB — сохранить байт. Переписывает содержимое AL в ячейку памяти с адресом ES: DI, после чего прибавляет единицу к DI, если флаг D опущен, и вычитает 1,если D поднят.

SHL – логический сдвиг операнда влево.

JNC – Переход, если перенос не установлен.

JA, JB – переход, если больше/меньше (без знака)

DATA SEGMENT

string\_1 DB "g#Xt@8@SYWpoXc@BRUylXlRPFxU1XInlcfJY6K56GteWrxvmoqmPFLrow9seh1cN" ;source string

string\_2 DB 64 dup(?)

string\_3 DB 64 dup(?)

bw1 DW 1101001010010000b ; bit strings

bw2 DW 1001101100110011b

bw3 DW 0100110010111010b

bw4 DW 1110111101110001b

DATA ENDS

STCK SEGMENT STACK

DB 256 DUP(?)

STCK ENDS

CODE SEGMENT 'code'

ASSUME CS:CODE,DS:DATA

start\_:

MOV AX,DATA

MOV DS,AX

MOV ES,AX

LEA SI,string\_1

LEA DI,string\_2

MOV BX,bw1

CLD ;clear flag df

begin\_1:

LODSB

SHL BX,1 ;move left on one bit

JNC next\_1 ;if bitC=0 - next

CMP AL,"A"

JB next\_1

CMP AL,"Z"

JA next\_1

STOSB

next\_1:

CMP SI,15 ;if(SI=15) load next part of bit string

JE load\_2

CMP SI,31 ;if(SI=31) load next part of bit string

JE load\_3

CMP SI,47 ;if(SI=47) load next part of bit string

JE load\_4

JMP no\_load

load\_2: ;load 2nd part of bit string

MOV BX,bw2

JMP no\_load

load\_3: ;load 3rd part of bit string

MOV BX,bw3

JMP no\_load

load\_4: ;load 4th part of bit string

MOV BX,bw4

no\_load:

CMP SI,64 ;if(i=64) exit from cycle

JE exit\_1

JMP begin\_1

exit\_1:

LEA SI,string\_1 ;cycle for 2nd result array

LEA DI,string\_3

ADD DI,63

begin\_2:

CLD

LODSB

CMP AH,"R"

JNE next\_2

CMP AL,"U"

JNE next\_2

JMP exit\_2

next\_2:

STD

STOSB

MOV AH,AL

CMP SI,64 ;if(i=64) -> exit

JE exit\_2

JMP begin\_2

exit\_2:

MOV AX,4C00h

INT 21h

CODE ENDS

END start\_