Сортировка по возрастанию 20 однобайтовых двоичных чисел. Вывод результата на экран. (пузырьком)

.model **small**

.**stack** 100h

**.data**

buf **db** 80

lnt **db** ?

string  **db** 80 dup**(**?**)**

.**code**

**start**:

**mov** **ax**,@**data**

**mov** **ds**,**ax**

**mov** **ah**,0ah

**lea** **dx**,buf

**int** 21h

**xor** **cx**,**cx**

@1: **mov** **cl**,lnt

**dec** **cl**

**lea** **si**,string

**xor** **bl**,**bl**

@2: **lodsb**

**cmp** **al**,**byte** ptr**[si]**

**jna** @3

**inc** **bl**

**xchg** **al**,**byte** ptr**[si]**

**mov** **byte** ptr**[si**-1**]**,**al**

@3: **loop** @2

**or** **bl**,**bl**

**jnz** @1

**mov** **byte** ptr**[si**+1**]**,'$'

**mov** **word** ptr**[**buf**]**,0d0ah

**mov** **ah**,9

**lea** **dx**,buf

**int** 21h

**mov** **ax**,4c00h

**int** 21h

end **start**

**шейкерная**

**.data**

H **dd** 0    *;верхняя граница неотсортированного массива*

L **dd** n-1    *;нижняя граница неотсортированного массива*

.**code**

**xor** **esi**,**esi**        *;нижняя граница неотсортированного массива*

**xor** **ebx**,**ebx**        *;флаг - были/не были перестановки в проходе*

**mov** **ecx**,L *;количество неотсортированных элементов снизу минус один*

a4:    **inc** **esi**

**call** Compare\_and\_Swapping    *;сравнение и обмен значений элементов*

**loop** a4                *;двигаемся вниз до границы массива*

**dec** **ebx**                *;проверяем были ли перестановки*

**jnz** exit        *;если перестановок не было - сортировка закончена*

**dec** L        *;уменьшаем количество неотсортированных элементов снизу*

**jz** exit    *;достигли границы массива*

**dec** **esi**        *;esi=L*

**mov** **ecx**,**esi**

**sub** **ecx**,H *;количество неотсортированных элементов сверху минус один*

**jecxz** exit    *;если граница снизу равна границе сверху - выходим*

a2:    **call** Compare\_Swapping        *;сравнение и обмен значений элементов*

**dec** **esi**

**loop** a2                *;двигаемся вверх до границы массива*

**dec** **ebx**                *;проверяем были ли перестановки*

**jnz** exit     *;если перестановок не было - заканчиваем сортировку*

**inc** H     *;уменьшаем количество неотсортированных элементов сверху*

**inc** **esi**        *;esi=H*

**mov** **ecx**,L

**sub** **ecx**,**esi**    *;если граница снизу больше, чем граница сверху – значит*

**ja** a4*;есть еще неотсортированные элементы - начинаем новый проход*

exit:    . . .

Compare\_Swapping proc*;сравнение и обмен значений соседних элементов*

**mov** **eax**,array**[esi**\*4**]**    *;получаем значение очередного элемента*

**cmp** array**[esi**\*4-4**]**,**eax**    *;сравниваем его с соседним элементом*

**jna** a3    *;если меньше или равен - идем к следующему элементу*

**seta** **bl**    *;если была перестановка - взводим флаг*

**xchg** **eax**,array**[esi**\*4-4**]**        *;меняем значения элементов местами*

**mov** array**[esi**\*4**]**,**eax**

a3:    **ret**

Compare\_Swapping endp