Отправить этот файл:

Выберите файл Файл не выбран

Send File

final.asm has been uploaded.

Close Assmebler code | Close Lexical analisation | Close Detected errors

row number	first word	second word	third word	fourth word	fifth word
0	.386				
1	mcrs	MACRO			
2	And	eax	,	ebx	
3	ENDM				
4	mcr	MACRO	parm		
5	And	eax	,	ebx	
6	ENDM				
7	data	segment			
8	stri	db	"big ukrainian boss's string"		
9	hexa	dw	3Eh		
10	bina	dd	01101101b		
11	deca	dd	21		
12	data	ends			
13	cod1	segment			
14	assume	cs:cod1	,	ds:data	
15	met2:				
16	Adc	ax	,	[ebx+eax+6]	
17	Neg	bh			
18	Neg	ebx			
19	Dec	ax			
20	Adc	ecx	,	[ebx+si+6]	
21	Adc	bx	,	bina	
22	Adc	ecx	,	stri	
23	Cmp	bx	,	ah	
24	Cmp	[ebx]	,	eax	
25	Cmp	deca	,	ah	
26	Cmp	bina	,	eax	
27	met1:				
28	Or	[eax+0110b]	,	ebx	
29	Or	bh	,	bl	
30	Or	al	,	dh	
31	Or	cx	,	dx	
32	Mov	[bp+si+5]	,	bl	
33	Mov	[eax+3Ah]	,	bx	
34	And	[bx+di+5]	,	8AF2h	
35	And	[ebx+6]	,	8AF22E4Bh	
36	Jc	met1			
37	And	ecx	,	ebx	
38	Jc	near	ptr	met1	
39	Jmp	far	ptr	met3	
40	cod1	ends			

row number	first word	second word	third word	fourth word	fifth word
41	cod2	segment			
42	assume	cs:cod2	,	ds:data	
43	met3:				
44	Mov	[bp+di+5]	,	bl	
45	Jmp	far	ptr	met2	
46	cod2	ends			
47	END				

.386

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
0	0	.386	Undefined lexem	undefined	255	0					

```
unexpected lexem ".386
" on line 0 on position 0
```

mcrs MACRO

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
1	0	mcrs	macro name	identifier	4	0					
1	1	MACRO	MACRO	directive	5	0					

And eax , ebx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
2	0	And	and	command	3	0					
2	1	eax	register32	register	3	0					
2	2	,	,	char	4	0					
2	3	ebx	register32	register	3	0					

ENDM

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
3	0	ENDM	ENDM	directive	4	0					

mcr MACRO parm

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
4	0	mcr	macro name	identifier	4	0					
4	1	MACRO	MACRO	directive	5	0					
4	2	parm	identifier	identifier	4	0					

And eax , ebx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
5	0	And	and	command	3	0					
5	1	eax	register32	register	3	0					
5	2	,	,	char	4	0					
5	3	ebx	register32	register	3	0					

ENDM

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
6	0	ENDM	ENDM	directive	4	0					

data segment

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
7	0	data	segment name	identifier	4	0					
7	1	segment	segment	directive	7	0					

stri db "big ukrainian boss's string"

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
8	0	stri	variable	identifier	4	0					
8	1	db	DB	directive	2	1					
8	2	"big ukrainian boss's string"	str const	const	255	0					

hexa dw 3Eh

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
9	0	hexa	variable	identifier	4	0					
9	1	dw	DW	directive	2	2					
9	2	3Eh	hex const	const	255	0					

bina dd 01101101b

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_le
10	0	bina	variable	identifier	4	0					
10	1	dd	DD	directive	2	4					
10	2	01101101b	bin const	const	255	0					

deca dd 21

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
11	0	deca	variable	identifier	4	0					
11	1	dd	DD	directive	2	4					
11	2	21	dec const	const	255	0					

data ends

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
12	0	data	segment name	identifier	4	0					
12	1	ends	ENDS	directive	4	0					

cod1 segment

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
13	0	cod1	segment name	identifier	4	0					
13	1	segment	segment	directive	7	0					

assume cs:cod1 , ds:data

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_
14	0	assume	ASSUME	directive	6	0					
14	1	cs:cod1	code segment	register:code segment identifier	255	0					
14	2	,	,	char	4	0					
14	3	ds:data	data segment	register:data segment identifier	255	0					

met2:

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
15	0	met2:	met	identifier	4	0					

Adc ax , [ebx+eax+6]

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	require
16	0	Adc	adc	command	3	0					
16	1	ax	register16	register	3	0					
16	2	,	,	char	4	0					
16	3	[ebx+eax+6]	pointer	memory	255	0					

Neg bh

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
17	0	Neg	neg	command	3	0					
17	1	bh	register08	register	3	0					

Neg ebx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
18	0	Neg	neg	command	3	0					
18	1	ebx	register32	register	3	0					

Dec ax

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
19	0	Dec	dec	command	3	0					
19	1	ax	register16	register	3	0					

Adc ecx , [ebx+si+6]

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_
20	0	Adc	adc	command	3	0					
20	1	ecx	register32	register	3	0					
20	2	,	,	char	4	0					
20	3	[ebx+si+6]	pointer	memory	255	0					

Adc bx , bina

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
21	0	Adc	adc	command	3	0					
21	1	bx	register16	register	3	0					
21	2	,	,	char	4	0					
21	3	bina	variable	identifier	4	0					

Adc ecx , stri

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
22	0	Adc	adc	command	3	0					
22	1	ecx	register32	register	3	0					
22	2	,	,	char	4	0					
22	3	stri	variable	identifier	4	0					

Cmp bx , ah

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
23	0	Cmp	cmp	command	3	0					
23	1	bx	register16	register	3	0					
23	2	,	,	char	4	0					
23	3	ah	register08	register	3	0					

```
Operand types must match on line 23
```

Cmp [ebx] , eax

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
24	0	Cmp	cmp	command	3	0					
24	1	[ebx]	pointer	memory	255	0					
24	2	,	,	char	4	0					
24	3	eax	register32	register	3	0					

Cmp deca , ah

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
25	0	Cmp	cmp	command	3	0					
25	1	deca	variable	identifier	4	0					
25	2	,	,	char	4	0					
25	3	ah	register08	register	3	0					

Cmp bina , eax

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
26	0	Cmp	cmp	command	3	0					
26	1	bina	variable	identifier	4	0					
26	2	,	,	char	4	0					
26	3	eax	register32	register	3	0					

met1:

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
27	0	met1:	met	identifier	4	0					

Or [eax+0110b] , ebx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	require
28	0	Or	or	command	2	0					
28	1	[eax+0110b]	pointer	memory	255	0					
28	2	,	,	char	4	0					
28	3	ebx	register32	register	3	0					

Or bh , bl

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
29	0	Or	or	command	2	0					
29	1	bh	register08	register	3	0					
29	2	,	,	char	4	0					
29	3	bl	register08	register	3	0					

Or al , dh

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
30	0	Or	or	command	2	0					
30	1	al	register08	register	3	0					
30	2	,	,	char	4	0					
30	3	dh	register08	register	3	0					

Or cx , dx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
31	0	Or	or	command	2	0					
31	1	cx	register16	register	3	0					
31	2	,	,	char	4	0					
31	3	dx	register16	register	3	0					

Mov [bp+si+5] , bl

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_l
32	0	Mov	mov	command	3	0					
32	1	[bp+si+5]	pointer	memory	255	0					
32	2	,	,	char	4	0					
32	3	bl	register08	register	3	0					

Mov [eax+3Ah] , bx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_
33	0	Mov	mov	command	3	0					
33	1	[eax+3Ah]	pointer	memory	255	0					
33	2	,	,	char	4	0					
33	3	bx	register16	register	3	0					

And [bx+di+5] , 8AF2h

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
34	0	And	and	command	3	0					
34	1	[bx+di+5]	pointer	memory	255	0					
34	2	,	,	char	4	0					
34	3	8AF2h	hex const	const	255	0					

And [ebx+6] , 8AF22E4Bh

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_l
35	0	And	and	command	3	0					
35	1	[ebx+6]	pointer	memory	255	0					
35	2	,	,	char	4	0					
35	3	8AF22E4Bh	hex const	const	255	0					

Jc met1

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexem
36	0	Jc	jc	command	2	0					
36	1	met1	identifier	identifier	4	0					

And ecx , ebx

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
37	0	And	and	command	3	0					
37	1	ecx	register32	register	3	0					
37	2	,	,	char	4	0					
37	3	ebx	register32	register	3	0					

Jc near ptr met1

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexem
38	0	Jc	jc	command	2	0					
38	1	near	near	directive	3	0					
38	2	ptr	identifier	identifier	4	0					
38	3	met1	identifier	identifier	4	0					

Jmp far ptr met3

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexem
39	0	Jmp	jmp	command	3	0					
39	1	far	far	directive	3	0					
39	2	ptr	identifier	identifier	4	0					
39	3	met3	identifier	identifier	4	0					

cod1 ends

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
40	0	cod1	segment name	identifier	4	0					
40	1	ends	ENDS	directive	4	0					

cod2 segment

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexe
41	0	cod2	segment name	identifier	4	0					
41	1	segment	segment	directive	7	0					

assume cs:cod2 , ds:data

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_
42	0	assume	ASSUME	directive	6	0					
42	1	cs:cod2	code segment	register:code segment identifier	255	0					
42	2	,	,	char	4	0					
42	3	ds:data	data segment	register:data segment identifier	255	0					

met3:

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
43	0	met3:	met	identifier	4	0					

Mov [bp+di+5] , bl

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_l
44	0	Mov	mov	command	3	0					
44	1	[bp+di+5]	pointer	memory	255	0					
44	2	,	,	char	4	0					
44	3	bl	register08	register	3	0					

Jmp far ptr met2

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexem
45	0	Jmp	jmp	command	3	0					
45	1	far	far	directive	3	0					
45	2	ptr	identifier	identifier	4	0					
45	3	met2	identifier	identifier	4	0					

cod2 ends

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
46	0	cod2	segment name	identifier	4	0					
46	1	ends	ENDS	directive	4	0					

END

row number	word number	code	name	type	max_length	weight	required_prev	required_next_elem	required_before	required_after	required_lexems
47	0	END	END	directive	3	0					

Error near line 0:

```
unexpected lexem ".386
```

```
" on line 0 on position 0
```

Error near line 23:

```
Operand types must match on line 23
```