

A

B

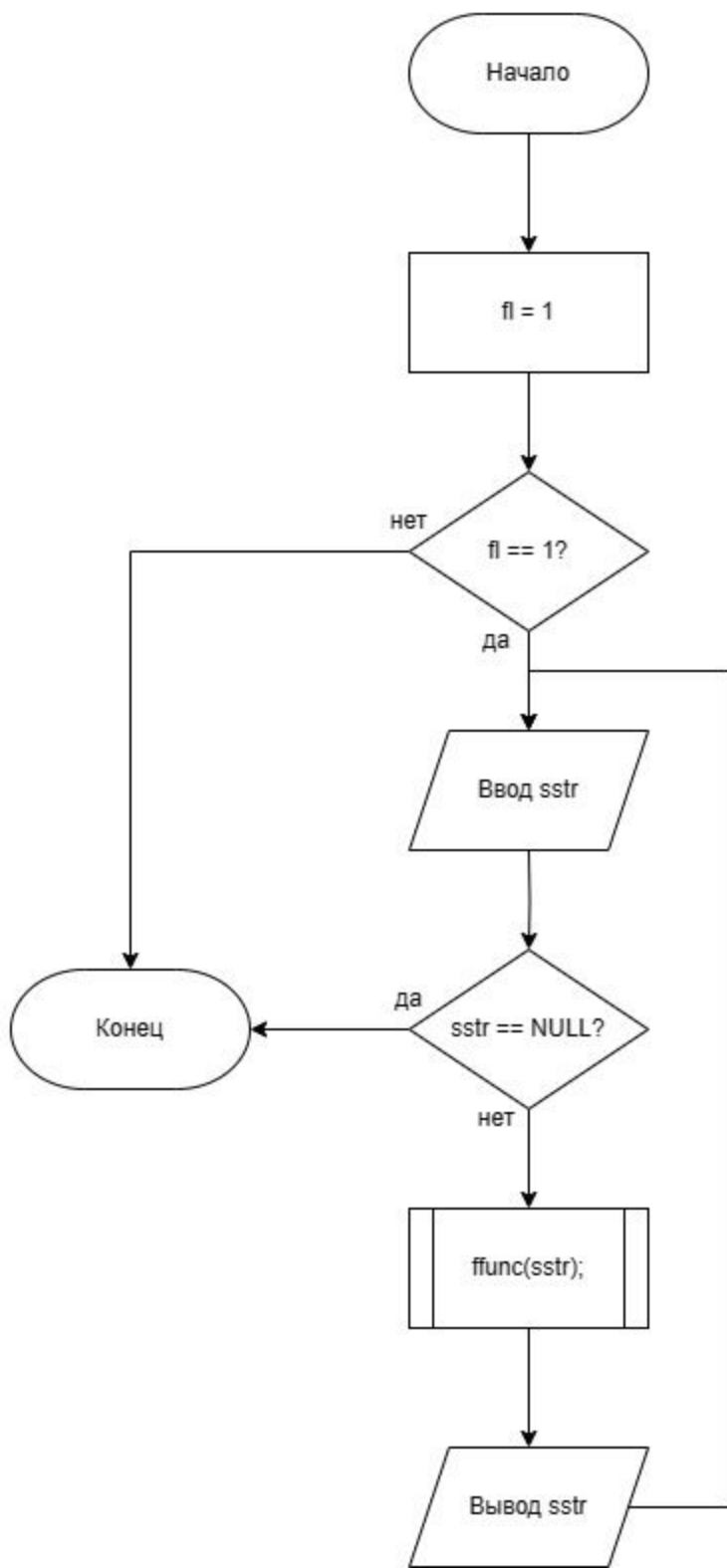
C

D

E

F

G



1 Блок-схема ffunc() 2

A

void ffunc(char *sstr)

B

len = strlen(sstr);
i = 0;

C

sstr[0] != ' '?
нет
да

D

i < len-1?
нет
да

E

sstr[i] = sstr[i+1]
i += 1

F

len -= 1

G

i = 0

H

Конец
нет
да
i < len?

I

sstr[i-1] == ' '?
нет
да

J

j = i

K

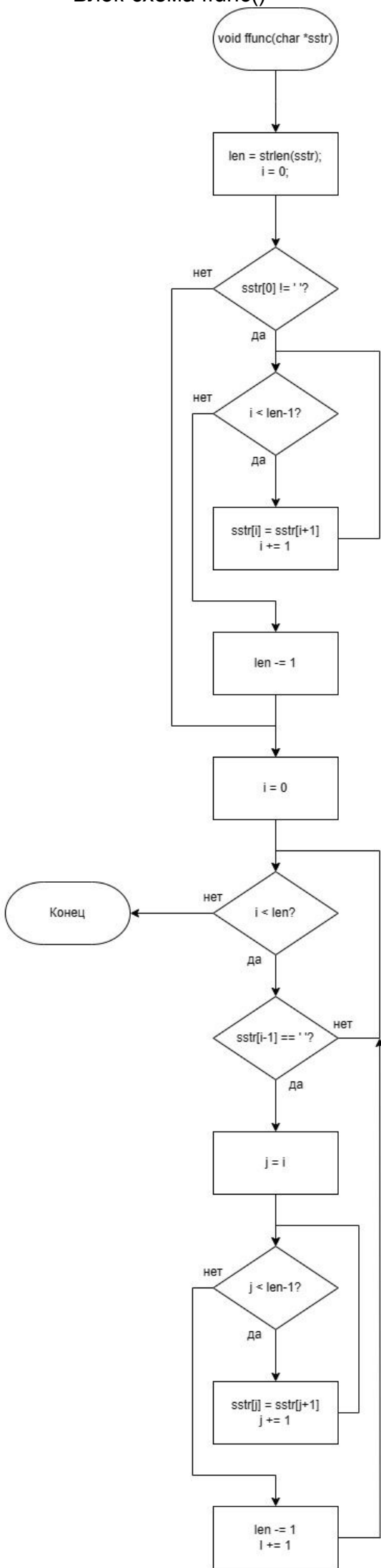
j < len-1?
нет
да

L

sstr[j] = sstr[j+1]
j += 1

M

len -= 1
i += 1



```
#include <stdlib.h>
#include <readline/readline.h>
#include "ffunc.h"

int main()
{
    char *sstr;
    int fl = 1;
    while (fl) { // fl = 1 always
        sstr = readline("Введите строку.\n");
        if (sstr == NULL) {
            printf("Завершение программы.");
            break;
        }

        ffunc(sstr);
        printf("%s\n", sstr);
        free(sstr);
    }

    return 0;
}
```

```
#include <string.h>
```

ffunc.c

```
#include "ffunc.h"
```

```
void ffunc(char *sstr)
```

```
{
```

```
    int i;
```

```
    int len = strlen(sstr);
```

```
    if (sstr[0] != ' ') {
```

```
        for (i = 0; i < len - 1; ++i) {
```

```
            sstr[i] = sstr[i+1];
```

```
        }
```

```
        sstr[len-1] = '\0';
```

```
        len--;
```

```
    }
```

```
    int j;
```

```
    for (i = 1; i < len; ++i) {
```

```
        if (sstr[i-1] == ' ') {
```

```
            for (j = i; j < len - 1; ++j) {
```

```
                sstr[j] = sstr[j+1];
```

```
            }
```

```
            sstr[len-1] = '\0';
```

```
            len--;
```

```
        }
```

```
    }
```

```
}
```

	A	B	Трассировка main()	
1	Оператор	Условие	sstr	Примечания
2	sstr = readline();		abc defg hi jklm	
3	sstr == NULL?	НЕТ		
4	funct(sstr)		bc efg l klm	
5	printf("%s\n",sstr);			
6	sstr = readline();		123 456 789	
7	sstr == NULL?	НЕТ		
8	funct(sstr)		23 56 89	
9	printf("%s\n",sstr);			
10	sstr = readline();		NULL	На вводе ctrl+D
11	sstr == NULL?	ДА		
12	printf("Конец");			

	A	B	C	D	E	F	G
1	Оператор	Условие	sstr	len	i	j	Примечания
2			abc de fghi				Принятый аргумент
3	len = strlen(sstr)			11			Трассировка ffunc()
4	s[0] != ''?	ДА					
5	i = 0				0		
6	i < len-1	ДА					
7	str[i] = str[i+1]		bbc de fghi				
8	i++				1		
9	i < len-1	ДА					
10	str[i] = str[i+1]		bcc de fghi				
11	i++				2		
12	i < len-1	ДА					
13	str[i] = str[i+1]		bc de fghi				
14	i++				3		
15	i < len-1	ДА					
16	str[i] = str[i+1]		bc dde fghi				
17	i++				4		
18	i < len-1	ДА					
19	str[i] = str[i+1]		bc dee fghi				
20	i++				5		
21	i < len-1	ДА					
22	str[i] = str[i+1]		bc de fghi				
23	i++				6		
24	i < len-1	ДА					
25	str[i] = str[i+1]		bc de ffgghi				
26	i++				7		
27	i < len-1	ДА					
28	str[i] = str[i+1]		bc de fgghi				
29	i++				8		
30	i < len-1	ДА					
31	str[i] = str[i+1]		bc de fghhi				
32	i++				9		
33	i < len-1	ДА					
34	str[i] = str[i+1]		bc de fghii				
35	i++				10		
36	i < len-1	НЕТ					
37	sstr[len-1] = '\0'		bc de fghi				

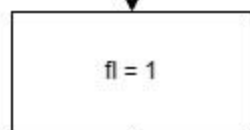
№	A	B	C	D	E	F	G
37	sstr[len-1] = '\0'		bc de fghi				
38	len--			10			
39	i = 1				1		
40	i < len?	ДА					
41	sstr[i-1] == ' '?	НЕТ					str[1-1] = 'b'
42	i++				2		
43	i < len?	ДА					
44	sstr[i-1] == ' '?	НЕТ					= 'c'
45	i++				3		
46	i < len?	ДА					
47	sstr[i-1] == ' '?	ДА					= ''
48	j = i					3	
49	j < len-1?	ДА					
50	str[j] = str[j+1]		bc ee fghi				
51	j ++					4	
52	j < len-1?	ДА					
53	str[j] = str[j+1]		bc e fghi				
54	j ++					5	
55	j < len-1?	ДА					
56	str[j] = str[j+1]		bc e ffghi				
57	j ++					6	
58	j < len-1?	ДА					
59	str[j] = str[j+1]		bc e fgghi				
60	j ++					7	
61	j < len-1?	ДА					
62	str[j] = str[j+1]		bc e fghhi				
63	j ++					8	
64	j < len-1?	ДА					
65	str[j] = str[j+1]		bc e fghii				
66	j ++					9	
67	j < len-1?	НЕТ					
68	sstr[len-1] = '\0'		bc e fghi				
69	len--			9			
70	i++				4		
71	i < len?	ДА					
72	sstr[i-1] == ' '?	НЕТ					= 'e'

	A	B	C	D	E	F	G
72	sstr[i-1] == ' '? i++	HET			5		= 'e'
74	i < len?	ДА					
75	sstr[i-1] == ' '? j = i	ДА				5	= ''
77	j < len-1?	ДА					
78	str[j] = str[j+1] j++		bc e gghi			6	
80	j < len-1?	ДА					
81	str[j] = str[j+1] j++		bc e ghhi			7	
83	j < len-1?	ДА					
84	str[j] = str[j+1] j++		bc e ghii			8	
86	j < len-1?	HET					
87	sstr[len-1] = '\0' len--		bc e ghi	8			
89	i++				6		
90	i < len?	ДА					
91	sstr[i-1] == ' '? i++	HET			7		= 'g'
93	i < len?	ДА					
94	sstr[i-1] == ' '? i++	HET			8		= 'h'
96	i < len?	HET					

A



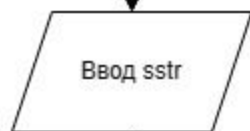
B



C



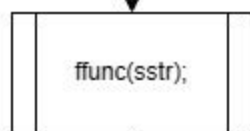
D



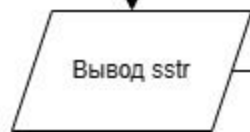
E



F



G



1

2

3

Блок-схема ffunc()

A

void ffunc(char *sstr)

B

```
strcpy(sstr,sstr);  
strcpy(sstr,sstr);  
strcpy(sstr,"");
```

C

token=strtok(sstr, " ");

D

token != NULL?

нет

да

E

tokLen = strlen(token)

F

tokLen % 2 == 0?

нет

да

G

```
strcat(sstr, token);  
strcat(sstr, " ");
```

H

```
token =  
strtok(NULL, " ");
```

token=strtok(strr, " ");

token != NULL?

нет

да

tokLen = strlen(token)

tokLen % 2 == 1?

нет

да

```
strcat(sstr, token);  
strcat(sstr, " ");
```

```
token =  
strtok(NULL, " ");
```

Конец

```
#include <stdlib.h>
#include <readline/readline.h>
#include "ffunc.h"

int main()
{
    char *sstr;
    int fl = 1;
    while (fl) { // fl = 1 always
        sstr = readline("Введите строку.\n");
        if (sstr == NULL) {
            printf("Завершение программы.");
            break;
        }

        ffunc(sstr);
        printf("%s\n", sstr);
        free(sstr);
    }

    return 0;
}
```

main.c

```
#ifndef FFUNC_H  
#define FFUNC_H  
  
void ffunc(char*);  
  
#endif
```

```
#include <stdlib.h>
#include <string.h>
#include "ffunc.h"
```

ffunc.c

```
void ffunc(char *sstr)
{
    int l = strlen(sstr);
    char sttr[l], strr[l];
    strcpy(sttr,sstr);
    strcpy(strr,sstr);

    strcpy(sstr,"");

    char *token = strtok(sttr, " ");
    while (token != NULL) {
        if (strlen(token) % 2 == 0) {
            strcat(sstr, token);
            strcat(sstr, " ");
        }
        token = strtok(NULL, " ");
    }

    token = strtok(strr, " ");
    while (token != NULL) {
        if (strlen(token) % 2 == 1) {
            strcat(sstr, token);
            strcat(sstr, " ");
        }
        token = strtok(NULL, " ");
    }
    // free(sttr);
    // free(strr);
}
```

Трассировка main()

	A	B	C	D
1	Оператор	Условие	sstr	Примечания
2	sstr = readline();		od1 e2 od3 e4	
3	sstr == NULL?	НЕТ		
4	funct(sstr)		e2 e4 od1 od3	
5	printf("%s\n",sstr);			
6	sstr = readline();		even_1 odd_2 even_3 even_4 odd_5 even_6 odd_7 odd_8 even_9	
7	sstr == NULL?	НЕТ		
8	funct(sstr)		even_1 even_3 even_4 even_6 even_9 odd_2 odd_5 odd_7 odd_8	
9	printf("%s\n",sstr);			
10	sstr = readline();		NULL	На вводе ctrl+D
11	sstr == NULL?	ДА		
12	printf("Конец");			

	A	B	C	D	E	F	G
1	Оператор	Условие	str1	str2	str3	token	Примечания
2			od1 e2 od3 e4				Принятые аргументы
3	strcpy(str2, str1)			od1 e2 od3 e4			Трассировка fflush()
4	strcpy(str3, str1)				od1 e2 od3 e4		
5	strcpy(str1, "")		""				
6	token = strtok(str2, " ")			od1\0e2 od3 e4		od1	
7	token != NULL	ДА					
8	strlen(token)%2==0?	НЕТ					
9	token=strtok(NULL, " ")			od1\0e2\0od3 e4		e2	
10	token != NULL	ДА					
11	strlen(token)%2==0?	ДА					
12	strcat(str1, token)		e2				
13	strcat(str1, " ")		e2_				_ обозначает пробел в конце
14	token=strtok(NULL, " ")			od1\0e2\0od3\0e4		od3	
15	token != NULL	ДА					
16	strlen(token)%2==0?	НЕТ					
17	token=strtok(NULL, " ")					e4	
18	token != NULL	ДА					
19	strlen(token)%2==0?	ДА					
20	strcat(str1, token)		e2 e4				
21	strcat(str1, " ")		e2 e4_				
22	token=strtok(NULL, " ")					NULL	
23	token != NULL	НЕТ					
24	token = strtok(str3, " ")				od1\0e2 od3 e4	od1	
25	token != NULL	ДА					
26	strlen(token)%2==1?	ДА					
27	strcat(str1, token)		e2 e4 od1				
28	strcat(str1, " ")		e2 e4 od1_				
29	token=strtok(NULL, " ")				od1\0e2\0od3 e4	e2	
30	token != NULL	ДА					
31	strlen(token)%2==1?	НЕТ					
32	token=strtok(NULL, " ")				od1\0e2\0od3\0e4	od3	
33	token != NULL	ДА					
34	strlen(token)%2==1?	ДА					
35	strcat(str1, token)		e2 e4 od1 od3				
36	strcat(str1, " ")		e2 e4 od1 od3_				
37	token=strtok(NULL, " ")					e4	
38	token != NULL	ДА					
39	strlen(token)%2==1?	НЕТ					
40	token=strtok(NULL, " ")					NULL	
41	token != NULL	НЕТ					
42							Конец программы