```
#include <stdlib.h>
#include <unistd.h>
int     ft_nb_char(char *str)
{
    int     i = 0;
    int     cnt = 0;
    int     first = 1;
    while (str[i])
    {
        if ((str[i] == ' ' || str[i] == '\t') && first == 1)
        {
            cnt++;
            first = 0;
        }
        else if (str[i] != ' ' && str[i] != '\t')
        {
            cnt++;
            first = 1;
        }
        i++;
    }
    return (cnt);
}

char    *trim_begin_end_space(char *str)
{
    char    *s;
    int     i;
    int     j;
    int     k;

    i = 0;
    k = 0;
    j = 0;
    while (str[j])
        j++;
    while (str[i] == ' ' || str[i] == '\t')
        i++;
    while (str[j - 1] == ' ' || str[i] == '\t')
        j--;
    s = (char*)malloc(sizeof(char) * (j - i + 1));
    if (s == NULL)
        return (NULL);
    while (k < j - i)
    {
        s[k] = str[i + k];
        k++;
    }
    s[k] = '\0';
    return (s);
}
```

```
char    *epur_str(char *str)
{
    int     t[] = { -1, 0 };
    int     first = 1;
    char    *s;
    str = trim_begin_end_space(str);
    s = (char*)malloc(sizeof(char) * (ft_nb_char(str) + 1));
    while (str[++t[0]])
    {
        if (str[t[0]] == ' ' || str[t[0]] == '\t')
        {
            if (first == 1)
                s[t[1]++] = str[t[0]];
            first = 0;
        }
        else
        {
            first = 1;
            s[t[1]++] = str[t[0]];
        }
    }
    free(str);
    s[t[1]] = '\0';
    return (s);
}
void    rostring(char *str)
{
    int     i = 0;
    int     j = 0;
    str = epur_str(str);
    if (str != NULL)
    {
        while (str[i] != ' ' && str[i] != '\t' && str[i])
            i++;
        i++;
        while (str[i + j])
        {
            write(1, &str[i + j], 1);
            j++;
        }
        if (str[i])
            write(1, " ", 1);
        j = -1;
        while (++j < i - 1)
            write(1, &str[j], 1);
        free(str);
    }
}
int         main(int argc, char **argv)
{
    if (argc > 1)
        rostring(argv[1]);
    write(1, "\n", 1);
    return (0);
}
```