

```

#include <unistd.h>
#include <stdlib.h>

void ft_putstr(char *str)
{
    int i;

    i = 0;
    while (str[i])
    {
        write(1, &str[i], 1);
        i++;
    }
}

```

```

int main(int argc, char **argv)
{
    char *mot;
    int i = 0;
    int d;
    int k = 0;
    mot = NULL;
    if (argc > 1)
    {
        while (argv[1][i] && (argv[1][i] == ' '
                               || argv[1][i] == '\t' || argv[1][i] == '\n'))
            i++;
        d = i;
        while (argv[1][i] && argv[1][i] != ' '
               && argv[1][i] != '\t' && argv[1][i] != '\n')
        {
            k++;
            i++;
        }
        mot = (char*)malloc(sizeof(char) * k + 1);
        i = 0;
        while (i < k)
        {
            mot[i] = argv[1][d + i];
            i++;
        }
        mot[k] = '\0';
        i = d + k;
        while (argv[1][i] && (argv[1][i] == ' '
                               || argv[1][i] == '\t' || argv[1][i] == '\n'))
            i++;
        d = 0;
        while (argv[1][i])
        {
            if (d == 1 && argv[1][i] != ' ' &&
                argv[1][i] != '\t' && argv[1][i] != '\n')
            {
                write(1, " ", 1);
                write(1, &argv[1][i], 1);
                d = 0;
            }
            else if (d == 0 && argv[1][i] != ' ' &&
                    argv[1][i] != '\t' && argv[1][i] != '\n')
                write(1, &argv[1][i], 1);
            else
                d = 1;
            i++;
        }
        if (i > k)
            write(1, " ", 1);
        ft_putstr(mot);
        free(mot);
    }
    write(1, "\n", 1);
    return (0);
}

```