

```

#include "4-5-check_mate-02.h"
size_t ft_strlen(char *s)
{
    size_t i = 0;
    while (s[i])
        i++;
    return (i);
}
int ft_opiece(char piece)
{
    if (piece == 'P' || piece == 'Q' || piece == 'B' || piece == 'R')
        return (1);
    return (0);
}
/**** Pawn *****/
int ft_pawn(char **board, int y, int x)
{
    if (y > 1)
    {
        if (board[y - 1][x - 1] == 'K')
            return (1);
        else if (board[y - 1][x + 1] == 'K')
            return (1);
    }
    return (0);
}
/*end-pawn*/
/**** Rook *****/
int ft_rook(char **board, int y, int x)
{
    int len, j;
    len = (int)ft_strlen(board[y]);
    j = x;
    while (++j < len && ft_opiece(board[y][j]) != 1) //Horizontal++
    {
        if (board[y][j] == 'K')
            return (1);
    }
    j = x;
    while (--j >= 0 && ft_opiece(board[y][j]) != 1) //Horizontal--
    {
        if (board[y][j] == 'K')
            return (1);
    }
    j = y;
    while (++j <= len && ft_opiece(board[j][x]) != 1) //Vertical--
    {
        if (board[j][x] == 'K')
            return (1);
    }
    j = y;
    while (--j >= 1 && ft_opiece(board[j][x]) != 1) //Vertical++
    {
        if (board[j][x] == 'K')
            return (1);
    }
    return (0);
}
/*end-rook*/

```

```

/**** Bishop *****/
int ft_bishop(char **board, int y, int x)
{
    int len;
    int i;
    int j;

    len = (int)ft_strlen(board[1]);
    i = y;
    j = x;
    while (++i <= len && ++j < len && ft_opiece(board[i][j]) != 1) //Down Right
    {
        if (board[i][j] == 'K')
            return (1);
    }
    i = y;
    j = x;
    while (--i >= 1 && --j >= 0 && ft_opiece(board[i][j]) != 1) //Down Left
    {
        if (board[i][j] == 'K')
            return (1);
    }
    i = y;
    j = x;
    while (--i >= 1 && ++j < len && ft_opiece(board[i][j]) != 1) //Up Right
    {
        if (board[i][j] == 'K')
            return (1);
    }
    i = y;
    j = x;
    while (--i >= 1 && --j >= 0 && ft_opiece(board[i][j]) != 1) //Up Left
    {
        if (board[i][j] == 'K')
            return (1);
    }
    return (0);
}
/*end-bishop*/

```

```

static int ft_checkmate(char **av)
{
    int i;
    int j;

    i = 1;
    while (av[i])
    {
        j = 0;
        while (av[i][j])
        {
            if (av[i][j] == 'R' && ft_rook(av, i, j) == 1)
                return (1);
            if (av[i][j] == 'P' && ft_pawn(av, i, j) == 1)
                return (1);
            if (av[i][j] == 'B' && ft_bishop(av, i, j) == 1)
                return (1);
            if (av[i][j] == 'Q' && (ft_bishop(av, i, j) == 1 || ft_rook(av, i, j) == 1))
                return (1);
            j++;
        }
        i++;
    }
    return (0);
}

int main (int ac, char **av)
{
    if (ac > 1 && ac == (int)(ft_strlen(av[1]) + 1))
    {
        int i;

        i = 1;
        while (av[i] != NULL)
        {
            if (((int)ft_strlen(av[i]) + 1) == ac)
                i++;
            else
            {
                write(1, "Fail\n", 5);
                return (0);
            }
        }
        if (ft_checkmate(av) == 1)
            write(1, "Success\n", 8);
        else
            write(1, "Fail\n", 5);
    }
    else if (ac > 1)
        write(1, "Fail\n", 5);
    else
        write(1, "\n", 1);
    return (0);
}

```