

exam 00

Assignment name : aff_a
Expected files : aff_a.c
Allowed functions: write

Write a program that takes a string, and displays the first 'a' character it encounters in it, followed by a newline. If there are no 'a' characters in the string, the program just writes a newline. If the number of parameters is not 1, the program displays 'a' followed by a newline.

Example:

```
$> ./aff_a "abc" | cat -e
a$
$> ./aff_a "dub0 a POIL" | cat -e
a$
$> ./aff_a "zz sent le poney" | cat -e
$
$> ./aff_a | cat -e
a$
```

```
int main(int ac, char **av)
{
    int i = 0;

    if (ac == 2)
    {
        while (av[1][i])
        {
            if (av[1][i] == 'a')
            {
                write(1, "a", 1);
                break;
            }
            i++;
        }
    }
    else
    {
        write(1, "a", 1);
    }
    write(1, "\n", 1);
    return (0);
}
```

Atama adı : aff_z
Beklenen dosyalar : aff_z.c
İzin verilen işlevler: write

Bir dize alan ve ilk 'z'yi görüntüleyen bir program yazın karakteriyle karşılaşır ve ardından bir satırsonu gelir. Eğer hiç yoksa Dizide 'z' karakteri varsa, program 'z'nin ardından şunları yazar bir satırbaşı ile. Parametre sayısı 1 değilse, program şunları görüntüler 'z' ve ardından bir satırbaşı.

Örnek:

```
$> ./aff_z "abc" | cat -e
z$
$> ./aff_z "dub0 a POIL" | cat -e
z$
$> ./aff_z "zaz sent le poney" | cat -e
z$
$> ./aff_z | cat -e
z$
*/
```

```
#include <unistd.h>
int main(int ac, char **av)
{
    int i = 0;
    if (ac == 2)
    {
        while (av[1][i])
        {
            if (av[1][i] == 'z')
            {
                write(1, "z\n", 2);
                return (0);
            }
            i++;
        }
        write(1, "z\n", 2);
    }
    else
    {
        write(1, "z\n", 2);
    }
    return (0);
}
```

Altarnetif Olarak Aslında Bu kadar uzatmanada gerek yok her türlü z yazıyor

```
int main (int ac, char **av)
{
    (void) ac;
    (void) av;
    write(1,"z\n",2);
    return (0);
}
```

Atama adı : aff_first_param
Beklenen dosyalar : aff_first_param.c
İzin verilen işlevler: write

Dizeleri argüman olarak alan ve ilk dizeyi görüntüleyen bir program yazın
argümanının ardından bir \n gelir.

Bağımsız değişken sayısı 1'den azsa, program \n görüntüler.

Örnek :

```
$> ./aff_first_param vincent mit "l'ane" dans un pre et "s'en" vint | cat -e
vincent$
$> ./aff_first_param "j'aime le fromage de chevre" | cat -e
j'aime le fromage de chevre$
$> ./aff_first_param
$
```

```
#include <unistd.h>
int main(int ac, char **av)
{
    int i = 0;
    if (ac > 1)
    {
        while (av[1][i])
        {
            write(1,&av[1][i],1);
            i++;
        }
    }
    write(1,"\n",1);
    return (0);
}
```

Normal'de C06'da ilk argümanı yazdıran programdan neredeyse farkı yok aynı kod yapısını kullanabilirsin fakat şurada önemli fark var koşul koymazsan, boş atınca parçalanma hatası veriyor çünkü ./a.out 'tan sonrakini alıyorsun boş gönderince hata veriyor eğer if'in (koşulun) dışında bir değer varsa alt satıra geçip sonlanıyor...

Atama adı : aff_last_param
Beklenen dosyalar : aff_last_param.c
İzin verilen işlevler: write

Dizeleri argüman olarak alan ve sonuncusunu görüntüleyen bir program yazın
argümanının ardından bir satırsonu gelir.

Bağımsız değişken sayısı 1'den azsa, program bir satırsonu görüntüler.

Örnekler:

```
$> ./aff_last_param "zaz" "mange" "des" "chats" | cat -e
chats$
$> ./aff_last_param "j'aime le savon" | cat -e
j'aime le savon$
$> ./aff_last_param
$
```

```

#include <unistd.h>
void ft_putchar (char *str)
{
    int i = 0;
    while (str[i] != '\0')
    {
        write(1,&str[i],1);
        i++;
    }
}

int main(int ac, char **av)
{
    int i = ac -1;
    if (ac > 1)
    {
        ft_putchar(av[i]);
        write(1,"\n",1);
        i--;
    }
    return (0);
}

```

Bu'da c06'de tersten argümanları yazdırma örneğine benziyor tek fark orada putstr kullanıyoruz, burada putchar kullanıyoruz bir tanesini yazdırmak istediğimizden kaynaklı