Exo 15:

```
int main (int argc, char **argv) {
    int fd
    assert (argv[1] != 0);
    fd = open (argv[1], O_RDONLY);

    tac_gourmand (fd)
    close (fd);
    return 0;
}

                    int
void tac_gourmand (*fd)
    char tab[1024]; int Pos;
    Pos = read (fd, tab, 1024)

    if (                    == 0) {
        write ( FD  &tab, Pos);
        return;
    }
    else tac_gourmand(fd);
    write ( FD  , &tab, Pos);
}
```

Exo 14

```
int FD = 1;   |-> STD_OUT
```

```
void tac_ Pseek (int fd) {
    char c;
    int Pus; int pos,
    pos = Pseek(fd, 0, SEEK_END);
    assert(pos != -1);
    while (1==1){
        if (Pusread(fd, &c, 1) != 0){
            assert(Pus != -1);
            write( FD    , &c, 1); }
        if(Pseek(fd, -2, seek_cur)== -1){
            break;
        }
    }
    return;
```

~~Exo 16 :~~

```
void tac_pread ( int fd) {
    char c;
    int pos; Pus;
    pos = Pseek(fd, 0, SEEK_END)
    assert (pos != -1);
    while ((pos--) >0) {
    Pus=pread(fd, &c, 1, pos);
        write (FD, &c, 1)
    }
    return;
}
```

Exo 16 :

```
void tac (int fd) {
    int pos, Pos, i ;
    char tab [TAILLE_TAMPON]; char c;
    pos = lseek( fd, 0, SEEK_END);
    while ( 1 == 1){
        Pos = pread(fd, tab, TAILLE_TAMPON, pos);
        pos -= Pos ;
        for( i = 0 ; i < Pos/2 ; i++) {
            c = tab [i];
            tab [i] = tab [Pos - i];
            tab [Pos - i] = c ;
        }
    }
    write ( FD, &tab, Pos );
    if (Pos == 0){ break ;}
}
return;
}
```

fonction inverse

Exo 17 :

```
void tail (int fd, int n) {
    int i, pos ; char c ; int m = 0 ;
    while ((pos = read (fd, &c, 1)) != 0) {
        if (c == '\n') { m ++ ; }
    lseek (fd, 0, SEEK_SET);
    while ((pos = read (fd, &c, 1)) != 0) {
        if (c == '\n') { m -- ; }
        if (m < n) {
            write (FD, &c, 1); }
    }
```

Correction :

```
#define TAILLE_TAMPON 1024 ;        static int fd, pos ;
void tail (                int n) {

    pos = lseek (fd, 0, SEEK_END);
    printlines (                n)
}

void print_lines (                int n) {
    char tab [TAILLE_TAMPONS] ; int readc, i, nbLines
    if (n == 0) return ;
    if (pos == 0) ____ ;
    readc = pread (fd, &tab, TAILLE_TAMPON, pos);
    pos -= readc ;
    for (                ; i = readc ; i > 0 ; i -- ) {
```

PDS

TD ⑤ do 1/70

Exo 17 suite :

```
(write (FD, tab+i, TAILLE_TAMPON-i)
;return;)
if (tab[i] == '\n') n--; if(n=0)
printf lines ( n - );
write (FD, &tab, TAILLE_TAMPON);
return;
}
```