PB

This exercise aims to use the functions of low-level I/O files, calls to the open, write and lseek system. They can use the standard library input functions, for example scanf(), **but** they cannot use the tandard library output functions (printf, putchar etc) (they can't even have these words in comments)

The purpose of the program is to search inside a binary file to find a hidden string.

You will be given a sequence of instructions that you have to follow to get the hidden string.

there are five types of instruction (i) go to the end/start of a file and rewind/advance n bytes, (ii) advanced/indentable n bytes of the current position and (iii) read n bytes for a buffer (iv) clear the buffer (v) stop/finish.

Input

The name of the binary file to open in read mode is passed in the first argument to the program (argv[1]).

If the opening of the file is an error, your program must stop and return zero

Either way the return value of the program should always be zero.

The input (standard input) is in a sequence of rows.

Each line contains a character and an integer separated by a space.

The sequence ends with a pair 's' 0(zero).

The characters and their meaning are

+ n : advance n bytes
- n : rewind n bytes
i n : go to the start of the file and advance n bytes

f n : go to the end of the fikle and go back n bytes
r n : read n bytes and append to a buffer.
l 0 : clear the buffer
s 0 : stop

Output

The output is the buffer written to the standard output using only the write command followed by a byte (which must be the new line).
To write a new line, the character , define a variable or constant and use write(), for example

```
const char barran='\n';
write(STDOUT_FILENO,&barran,1);
```

Constraints

The buffer has a max of 100 bytes therefore the output string will be at most 99 chars

In this exercise you can not use printf - the word printf – the word can not appear anywhere in your program - instead of the printf use

write (STDOUT FILENO,..)

to write to the screen.

Invalid Submission = uses the standard library .. printf.

You can see if the word printf is in your program by using the grep command.

Samples

Consider the exactly 18-byte file .bin that contains the next string (no end ofline)

"Ford was president"
Note that it is 18 bytes since it is 16 characters (a-z) and 2 spaces.

You should create this file to perform tests using echo -n and check it's ok with ls -l and octal dump.

```
echo -n "Ford was president" > test.bin
ls -l test.bin
od -c test.bin
```

Consider the inputs in the input1 and input2 files as below.
You should run your program with your tests as follows:

```
./a.out test.bin < input1
./a.out test.bin < input2
```

Sample Input 1

```
i 9
r 1
- 4
r 2
+ 4
r 1
- 8
r 1
i 1
r 3
s 0
```

Sample Output 1
password

Sample Input 2
i 2
i 0
r 3
f 2
- 2
r 4
s 0

Sample Output 2
Fordent