Tasks

1. Write a grep command that will look for single-line comments in the both files (log3.py and log3.c).

grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'

Please note that that ' character is different from the ‘ character. The character ' will work, but the character ‘ will not.

```
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'
log3.c:    // Calculate log base 3 using natural logarithm (log(N) / log(3))
log3.c:    // Check if result is close to an integer to determine if N is a power of 3
log3.py:# this is a python comment
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"
log3.py:import math
log3.c:#include <stdbool.h>
log3.c:#include <math.h>
log3.c:#include <limits.h>
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*if ' log3.py log3.c
log3.py:if N==1:
log3.py:if powers_of_three[-1] == N:
log3.c:    if (N <= 0) return -1; // Logarithm undefined for non-positive numbers
log3.c:    if (fabs(result - round(result)) < 1e-10) {  // tolerance to handle floating-point precision
log3.c:    if (N < 1) {
log3.c:    if (powers[last] == N) printf("%d\n", log_base_3(N));
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='
log3.py:N = int(input())
log3.py:found = True
log3.py:powers_of_three = [3]
log3.c:    double result = log(N) / log(3);
log3.c:    int last = 0;
log3.c:    powers[last] = 1;
log3.c:        powers[last+1] = powers[last] * 3;
log3.c:        N/=powers[last];
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c
log3.c:    double result = log(N) / log(3);
log3.c:    long long N;
log3.c:    long long powers[MAX+1];
log3.c:    int last = 0;
(base) guoji@dyn172-30-66-177 Lab11 %
```

2. Write a grep command that will look for lines that import/include libraries in both files (log3.py and log3.c). These are lines that start with “import” in Python or “#include” in C. For Python, any import statement should be matched. For C, however, only lines that include a standard library header file should be matched. Therefore, the line “#include “utils.h”” should not be matched.

grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"

Please note that " is different from “ . Only the " will work. The “ character will not work.

```
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'
log3.c:    // Calculate log base 3 using natural logarithm (log(N) / log(3))
log3.c:    // Check if result is close to an integer to determine if N is a power of 3
log3.py:# this is a python comment
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"
log3.py:import math
log3.c:#include <stdbool.h>
log3.c:#include <math.h>
log3.c:#include <limits.h>
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*if ' log3.py log3.c
log3.py:if N==1:
log3.py:if powers_of_three[-1] == N:
log3.c:    if (N <= 0) return -1; // Logarithm undefined for non-positive numbers
log3.c:    if (fabs(result - round(result)) < 1e-10) {  // tolerance to handle floating-point precision
log3.c:    if (N < 1) {
log3.c:    if (powers[last] == N) printf("%d\n", log_base_3(N));
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='
log3.py:N = int(input())
log3.py:found = True
log3.py:powers_of_three = [3]
log3.c:    double result = log(N) / log(3);
log3.c:    int last = 0;
log3.c:    powers[last] = 1;
log3.c:        powers[last+1] = powers[last] * 3;
log3.c:        N/=powers[last];
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c
log3.c:    double result = log(N) / log(3);
log3.c:    long long N;
log3.c:    long long powers[MAX+1];
log3.c:    int last = 0;
(base) guoji@dyn172-30-66-177 Lab11 %
```

3. Write a grep command that will match lines that contain an if statement in both files (log3.py and log3.c).

grep '^[[:space:]]*if ' log3.py log3.c


```
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'
log3.c:    // Calculate log base 3 using natural logarithm (log(N) / log(3))
log3.c:    // Check if result is close to an integer to determine if N is a power of 3
log3.py:# this is a python comment
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"
log3.py:import math
log3.c:#include <stdbool.h>
log3.c:#include <math.h>
log3.c:#include <limits.h>
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*if ' log3.py log3.c
log3.py:if N==1:
log3.py:if powers_of_three[-1] == N:
log3.c:    if (N <= 0) return -1; // Logarithm undefined for non-positive numbers
log3.c:    if (fabs(result - round(result)) < 1e-10) {  // tolerance to handle floating-point precision
log3.c:    if (N < 1) {
log3.c:    if (powers[last] == N) printf("%d\n", log_base_3(N));
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='
log3.py:N = int(input())
log3.py:found = True
log3.py:powers_of_three = [3]
log3.c:    double result = log(N) / log(3);
log3.c:    int last = 0;
log3.c:    powers[last] = 1;
log3.c:        powers[last+1] = powers[last] * 3;
log3.c:        N/=powers[last];
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c
log3.c:    double result = log(N) / log(3);
log3.c:    long long N;
log3.c:    long long powers[MAX+1];
log3.c:    int last = 0;
(base) guoji@dyn172-30-66-177 Lab11 %
```

4. Write a grep command that will match lines that contain any assignment operator (e.g. =, /=, +=, etc) but not relational operators (e.g. ==, <=, etc) in both files (log3.py and log3.c).

grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='

Once again, please note the difference between ' and ’.

```
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'
log3.c:    // Calculate log base 3 using natural logarithm (log(N) / log(3))
log3.c:    // Check if result is close to an integer to determine if N is a power of 3
log3.py:# this is a python comment
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"
log3.py:import math
log3.c:#include <stdbool.h>
log3.c:#include <math.h>
log3.c:#include <limits.h>
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*if ' log3.py log3.c
log3.py:if N==1:
log3.py:if powers_of_three[-1] == N:
log3.c:    if (N <= 0) return -1; // Logarithm undefined for non-positive numbers
log3.c:    if (fabs(result - round(result)) < 1e-10) {  // tolerance to handle floating-point precision
log3.c:    if (N < 1) {
log3.c:    if (powers[last] == N) printf("%d\n", log_base_3(N));
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='
log3.py:N = int(input())
log3.py:found = True
log3.py:powers_of_three = [3]
log3.c:    double result = log(N) / log(3);
log3.c:    int last = 0;
log3.c:    powers[last] = 1;
log3.c:        powers[last+1] = powers[last] * 3;
log3.c:        N/=powers[last];
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c
log3.c:    double result = log(N) / log(3);
log3.c:    long long N;
log3.c:    long long powers[MAX+1];
log3.c:    int last = 0;
(base) guoji@dyn172-30-66-177 Lab11 %
```

5. Write a grep command that matches a variable declaration of the types (int, double, or long long) in the C file. Arrays declaration should also be matched, so the line: long long powers[MAX+1]; should be matched.

grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c

```
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*#\|^[[:space:]]*//' log3.c log3.py | grep -v '[[:space:]]*\(include\|define\)'
log3.c:    // Calculate log base 3 using natural logarithm (log(N) / log(3))
log3.c:    // Check if result is close to an integer to determine if N is a power of 3
log3.py:# this is a python comment
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*import \|^[[:space:]]*#*include' log3.py log3.c | grep -v \"
log3.py:import math
log3.c:#include <stdbool.h>
log3.c:#include <math.h>
log3.c:#include <limits.h>
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*if ' log3.py log3.c
log3.py:if N==1:
log3.py:if powers_of_three[-1] == N:
log3.c:    if (N <= 0) return -1; // Logarithm undefined for non-positive numbers
log3.c:    if (fabs(result - round(result)) < 1e-10) {  // tolerance to handle floating-point precision
log3.c:    if (N < 1) {
log3.c:    if (powers[last] == N) printf("%d\n", log_base_3(N));
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '[=]' log3.py log3.c | grep -v ' [>!<]' | grep -v '=='
log3.py:N = int(input())
log3.py:found = True
log3.py:powers_of_three = [3]
log3.c:    double result = log(N) / log(3);
log3.c:    int last = 0;
log3.c:    powers[last] = 1;
log3.c:        powers[last+1] = powers[last] * 3;
log3.c:        N/=powers[last];
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 %
(base) guoji@dyn172-30-66-177 Lab11 % grep '^[[:space:]]*\(long \| int \| double\)' log3.py log3.c
log3.c:    double result = log(N) / log(3);
log3.c:    long long N;
log3.c:    long long powers[MAX+1];
log3.c:    int last = 0;
(base) guoji@dyn172-30-66-177 Lab11 %
```