Building a Smart IDE: Identifying comments

Jack wants to build an IDE on his own. Help him build a feature which identifies the comments, in the source code of computer programs. Assume, that the programs are written either in C, C++ or Java. The commenting conventions are displayed here, for your convenience. At this point of time you only need to handle simple and common kinds of comments. You don't need to handle nested comments, or multi-line comments inside single comments or single-comments inside multi-line comments.

Your task is to write a program, which accepts as input, a C or C++ or Java program and outputs only the comments from those programs. Your program will be tested on source codes of not more than 200 lines.

Comments in C, C++ and Java programs

Single Line Comments:

```
// this is a single line comment
x = 1; // a single line comment after code
```

Please note that in the real world, some C compilers do not necessarily support the above kind of comment(s) but for the purpose of this problem let's just assume that the compiler which will be used here will accept these kind of comments.

Multi Line Comments:

```
/* This is one way of writing comments */
/* This is a multiline
   comment. These can often
   be useful*/
```

Input Format

Each test case will be the source code of a program written in C or C++ or Java.

Output Format

From the program given to you, remove everything other than the comments.

Sample Input #00

```
 // my  program in C++

#include <iostream>
/** playing around in
a new programming language **/
using namespace std;

int main ()
{
  cout << "Hello World";
  cout << "I'm a C++ program"; //use cout
  return 0;
}
```

Sample Output #00

```
// my  program in C++
/** playing around in
a new programming language **/
//use cout
```

Sample Input #01

```
 /*This is a program to calculate area of a circle after getting the radius as input from the user*/
#include<stdio.h>
int main()
{
   double radius,area;//variables for storing radius and area
   printf("Enter the radius of the circle whose area is to be calculated\n");
   scanf("%lf",&radius);//entering the value for radius of the circle as float data type
   area=(22.0/7.0)*pow(radius,2);//Mathematical function pow is used to calculate square of radius
   printf("The area of the circle is %lf",area);//displaying the results
   getch();
}
/*A test run for the program was carried out and following output was observed
If 50 is the radius of the circle whose area is to be calculated
The area of the circle is 7857.1429*/
```

Sample Output #01

```
/*This is a program to calculate area of a circle after getting the radius as input from the user*/
//variables for storing radius and area
//entering the value for radius of the circle as float data type
//Mathematical function pow is used to calculate square of radius
//displaying the results
/*A test run for the program was carried out and following output was observed
If 50 is the radius of the circle whose area is to be calculated
The area of the circle is 7857.1429*/
```

Precautions

Do not add any leading or trailing spaces. Remove any leading white space before comments, including from all lines of a multi-line comment. Do not alter the line structure of multi-line comments except to remove leading whitespace. i.e. Do not collapse them into one line.