Problem 2254: Design Video Sharing Platform

Problem Information

Difficulty: Hard
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You have a video sharing platform where users can upload and delete videos. Each

video

is a

string

of digits, where the

i

th

digit of the string represents the content of the video at minute

i

. For example, the first digit represents the content at minute

0

in the video, the second digit represents the content at minute

1

in the video, and so on. Viewers of videos can also like and dislike videos. Internally, the platform keeps track of the

number of views, likes, and dislikes

on each video.

When a video is uploaded, it is associated with the smallest available integer

videoId

starting from

0

. Once a video is deleted, the

videoId

associated with that video can be reused for another video.

Implement the

VideoSharingPlatform

class:

VideoSharingPlatform()

Initializes the object.

int upload(String video)

The user uploads a

video

. Return the

videoId

associated with the video.

void remove(int videoId)

If there is a video associated with

videoId

, remove the video.

String watch(int videoId, int startMinute, int endMinute)

If there is a video associated with

videoId

, increase the number of views on the video by

1

and return the substring of the video string starting at

startMinute

and ending at

min(endMinute, video.length - 1

)

(

inclusive

). Otherwise, return

"-1"

.

void like(int videoId)

Increases the number of likes on the video associated with

videoId

by

1

if there is a video associated with

videoId

.

void dislike(int videoId)

Increases the number of dislikes on the video associated with

videoId

by

1

if there is a video associated with

videoId

.

int[] getLikesAndDislikes(int videoId)

Return a

0-indexed

integer array

values

of length

2

where

values[0]

is the number of likes and

values[1]

is the number of dislikes on the video associated with

videoId

. If there is no video associated with

videoId

, return

[-1]

.

int getViews(int videoId)

Return the number of views on the video associated with

videoId

, if there is no video associated with

videoId

, return

-1

.

Example 1:

Input

["VideoSharingPlatform", "upload", "upload", "remove", "remove", "upload", "watch", "watch", "like", "dislike", "dislike", "getLikesAndDislikes", "getViews"] [[], ["123"], ["456"], [4], [0], ["789"], [1, 0, 5], [1, 0, 1], [1], [1], [1], [1], [1]]

Output

[null, 0, 1, null, null, 0, "456", "45", null, null, null, [1, 2], 2]

Explanation

VideoSharingPlatform videoSharingPlatform = new VideoSharingPlatform(); videoSharingPlatform.upload("123"); // The smallest available videoId is 0, so return 0. videoSharingPlatform.upload("456"); // The smallest available

videoId

is 1, so return 1. videoSharingPlatform.remove(4); // There is no video associated with videoId 4, so do nothing. videoSharingPlatform.remove(0); // Remove the video associated with videoId 0. videoSharingPlatform.upload("789"); // Since the video associated with videoId 0 was deleted, // 0 is the smallest available

videoId

, so return 0. videoSharingPlatform.watch(1, 0, 5); // The video associated with videoId 1 is "456". // The video from minute 0 to min(5, 3 - 1) = 2 is "456", so return "456". videoSharingPlatform.watch(1, 0, 1); // The video associated with videoId 1 is "456". // The video from minute 0 to min(1, 3 - 1) = 1 is "45", so return "45". videoSharingPlatform.like(1); //

Increase the number of likes on the video associated with videoId 1. videoSharingPlatform.dislike(1); // Increase the number of dislikes on the video associated with videoId 1. videoSharingPlatform.dislike(1); // Increase the number of dislikes on the video associated with videoId 1. videoSharingPlatform.getLikesAndDislikes(1); // There is 1 like and 2 dislikes on the video associated with videoId 1, so return [1, 2]. videoSharingPlatform.getViews(1); // The video associated with videoId 1 has 2 views, so return 2.

Example 2:

Input

["VideoSharingPlatform", "remove", "watch", "like", "dislike", "getLikesAndDislikes", "getViews"] [[], [0], [0, 0, 1], [0], [0], [0], [0]]

Output

[null, null, "-1", null, null, [-1], -1]

Explanation

VideoSharingPlatform videoSharingPlatform = new VideoSharingPlatform(); videoSharingPlatform.remove(0); // There is no video associated with videoId 0, so do nothing. videoSharingPlatform.watch(0, 0, 1); // There is no video associated with videoId 0, so return "-1". videoSharingPlatform.like(0); // There is no video associated with videoId 0, so do nothing. videoSharingPlatform.dislike(0); // There is no video associated with videoId 0, so do nothing. videoSharingPlatform.getLikesAndDislikes(0); // There is no video associated with videoId 0, so return [-1]. videoSharingPlatform.getViews(0); // There is no video associated with videoId 0, so return -1.

Constraints:

1 <= video.length <= 10

5

The sum of

video.length

over all calls to

upload

does not exceed

10

5

video

consists of digits.

0 <= videoId <= 10

5

0 <= startMinute < endMinute < 10

5

startMinute < video.length

The sum of

endMinute - startMinute

over all calls to

watch

does not exceed

10

5

.

At most

10

5

calls

in total

will be made to all functions.

Code Snippets

C++:

```
class VideoSharingPlatform {
public:
VideoSharingPlatform() {

}

int upload(string video) {

}

void remove(int videoId) {

}

string watch(int videoId, int startMinute, int endMinute) {

}

void like(int videoId) {

}

void dislike(int videoId) {
```

```
}

vector<int> getLikesAndDislikes(int videoId) {

}

int getViews(int videoId) {

}
};

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform* obj = new VideoSharingPlatform();
* int param_1 = obj->upload(video);
* obj->remove(videoId);
* string param_3 = obj->watch(videoId,startMinute,endMinute);
* obj->like(videoId);
* obj->dislike(videoId);
* vector<int> param_6 = obj->getLikesAndDislikes(videoId);
* int param_7 = obj->getViews(videoId);
*/
```

Java:

```
class VideoSharingPlatform {

public VideoSharingPlatform() {

}

public int upload(String video) {

}

public void remove(int videoId) {

}

public String watch(int videoId, int startMinute, int endMinute) {

}
```

```
public void like(int videoId) {

}

public void dislike(int videoId) {

}

public int[] getLikesAndDislikes(int videoId) {

}

public int getViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = new VideoSharingPlatform();
* int param_1 = obj.upload(video);
* obj.remove(videoId);
* String param_3 = obj.watch(videoId,startMinute,endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* int[] param_6 = obj.getLikesAndDislikes(videoId);
* int param_7 = obj.getViews(videoId);
*/
```

Python3:

```
class VideoSharingPlatform:

def __init__(self):


def upload(self, video: str) -> int:


def remove(self, videoId: int) -> None:
```

```
def watch(self, videoId: int, startMinute: int, endMinute: int) -> str:


def like(self, videoId: int) -> None:


def dislike(self, videoId: int) -> None:


def getLikesAndDislikes(self, videoId: int) -> List[int]:


def getViews(self, videoId: int) -> int:



# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform()
# param_1 = obj.upload(video)
# obj.remove(videoId)
# param_3 = obj.watch(videoId,startMinute,endMinute)
# obj.like(videoId)
# obj.dislike(videoId)
# param_6 = obj.getLikesAndDislikes(videoId)
# param_7 = obj.getViews(videoId)
```

Python:

```
class VideoSharingPlatform(object):

def __init__(self):


def upload(self, video):
"""
:type video: str
:rtype: int
"""


def remove(self, videoId):
```

```
"""
:type videoId: int
:rtype: None
"""


def watch(self, videoId, startMinute, endMinute):
"""
:type videoId: int
:type startMinute: int
:type endMinute: int
:rtype: str
"""


def like(self, videoId):
"""
:type videoId: int
:rtype: None
"""


def dislike(self, videoId):
"""
:type videoId: int
:rtype: None
"""


def getLikesAndDislikes(self, videoId):
"""
:type videoId: int
:rtype: List[int]
"""


def getViews(self, videoId):
"""
:type videoId: int
:rtype: int
"""
```

```
# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform()
# param_1 = obj.upload(video)
# obj.remove(videoId)
# param_3 = obj.watch(videoId,startMinute,endMinute)
# obj.like(videoId)
# obj.dislike(videoId)
# param_6 = obj.getLikesAndDislikes(videoId)
# param_7 = obj.getViews(videoId)
```

JavaScript:

```
var VideoSharingPlatform = function() {

};

/**
* @param {string} video
* @return {number}
*/
VideoSharingPlatform.prototype.upload = function(video) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.remove = function(videoId) {

};

/**
* @param {number} videoId
* @param {number} startMinute
* @param {number} endMinute
* @return {string}
*/
VideoSharingPlatform.prototype.watch = function(videoId, startMinute,
```

```
endMinute) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.like = function(videoId) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.dislike = function(videoId) {

};

/**
* @param {number} videoId
* @return {number[]}
*/
VideoSharingPlatform.prototype.getLikesAndDislikes = function(videoId) {

};

/**
* @param {number} videoId
* @return {number}
*/
VideoSharingPlatform.prototype.getViews = function(videoId) {

};

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = new VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
```

```
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

TypeScript:

```
class VideoSharingPlatform {
constructor() {

}

upload(video: string): number {

}

remove(videoId: number): void {

}

watch(videoId: number, startMinute: number, endMinute: number): string {

}

like(videoId: number): void {

}

dislike(videoId: number): void {

}

getLikesAndDislikes(videoId: number): number[] {

}

getViews(videoId: number): number {

}
}

/**
```

```
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = new VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

C#:

```
public class VideoSharingPlatform {

public VideoSharingPlatform() {

}

public int Upload(string video) {

}

public void Remove(int videoId) {

}

public string Watch(int videoId, int startMinute, int endMinute) {

}

public void Like(int videoId) {

}

public void Dislike(int videoId) {

}

public int[] GetLikesAndDislikes(int videoId) {

}
```

```
public int GetViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = new VideoSharingPlatform();
* int param_1 = obj.Upload(video);
* obj.Remove(videoId);
* string param_3 = obj.Watch(videoId,startMinute,endMinute);
* obj.Like(videoId);
* obj.Dislike(videoId);
* int[] param_6 = obj.GetLikesAndDislikes(videoId);
* int param_7 = obj.GetViews(videoId);
*/
```

C:

```
typedef struct {

} VideoSharingPlatform;


VideoSharingPlatform* videoSharingPlatformCreate() {

}

int videoSharingPlatformUpload(VideoSharingPlatform* obj, char* video) {

}

void videoSharingPlatformRemove(VideoSharingPlatform* obj, int videoId) {

}

char* videoSharingPlatformWatch(VideoSharingPlatform* obj, int videoId, int
startMinute, int endMinute) {
```

```
}

void videoSharingPlatformLike(VideoSharingPlatform* obj, int videoId) {

}

void videoSharingPlatformDislike(VideoSharingPlatform* obj, int videoId) {

}

int* videoSharingPlatformGetLikesAndDislikes(VideoSharingPlatform* obj, int
videoId, int* retSize) {

}

int videoSharingPlatformGetViews(VideoSharingPlatform* obj, int videoId) {

}

void videoSharingPlatformFree(VideoSharingPlatform* obj) {

}

/**
* Your VideoSharingPlatform struct will be instantiated and called as such:
* VideoSharingPlatform* obj = videoSharingPlatformCreate();
* int param_1 = videoSharingPlatformUpload(obj, video);

* videoSharingPlatformRemove(obj, videoId);

* char* param_3 = videoSharingPlatformWatch(obj, videoId, startMinute,
endMinute);

* videoSharingPlatformLike(obj, videoId);

* videoSharingPlatformDislike(obj, videoId);

* int* param_6 = videoSharingPlatformGetLikesAndDislikes(obj, videoId,
retSize);

* int param_7 = videoSharingPlatformGetViews(obj, videoId);
```

```
* videoSharingPlatformFree(obj);
*/
```

Go:

```
type VideoSharingPlatform struct {

}


func Constructor() VideoSharingPlatform {

}


func (this *VideoSharingPlatform) Upload(video string) int {

}


func (this *VideoSharingPlatform) Remove(videoId int) {

}


func (this *VideoSharingPlatform) Watch(videoId int, startMinute int,
endMinute int) string {

}


func (this *VideoSharingPlatform) Like(videoId int) {

}


func (this *VideoSharingPlatform) Dislike(videoId int) {

}
```

```
func (this *VideoSharingPlatform) GetLikesAndDislikes(videoId int) []int {

}


func (this *VideoSharingPlatform) GetViews(videoId int) int {

}


/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* obj := Constructor();
* param_1 := obj.Upload(video);
* obj.Remove(videoId);
* param_3 := obj.Watch(videoId,startMinute,endMinute);
* obj.Like(videoId);
* obj.Dislike(videoId);
* param_6 := obj.GetLikesAndDislikes(videoId);
* param_7 := obj.GetViews(videoId);
*/
```

Kotlin:

```
class VideoSharingPlatform() {

fun upload(video: String): Int {

}

fun remove(videoId: Int) {

}

fun watch(videoId: Int, startMinute: Int, endMinute: Int): String {

}

fun like(videoId: Int) {

}
```

```
fun dislike(videoId: Int) {

}

fun getLikesAndDislikes(videoId: Int): IntArray {

}

fun getViews(videoId: Int): Int {

}

}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

Swift:

```
class VideoSharingPlatform {

init() {

}

func upload(_ video: String) -> Int {

}

func remove(_ videoId: Int) {

}
```

```
func watch(_ videoId: Int, _ startMinute: Int, _ endMinute: Int) -> String {

}

func like(_ videoId: Int) {

}

func dislike(_ videoId: Int) {

}

func getLikesAndDislikes(_ videoId: Int) -> [Int] {

}

func getViews(_ videoId: Int) -> Int {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* let obj = VideoSharingPlatform()
* let ret_1: Int = obj.upload(video)
* obj.remove(videoId)
* let ret_3: String = obj.watch(videoId, startMinute, endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* let ret_6: [Int] = obj.getLikesAndDislikes(videoId)
* let ret_7: Int = obj.getViews(videoId)
*/
```

Rust:

```
struct VideoSharingPlatform {

}


/**
```

```
* `&self` means the method takes an immutable reference.
* If you need a mutable reference, change it to `&mut self` instead.
*/
impl VideoSharingPlatform {

fn new() -> Self {

}

fn upload(&self, video: String) -> i32 {

}

fn remove(&self, video_id: i32) {

}

fn watch(&self, video_id: i32, start_minute: i32, end_minute: i32) -> String
{

}

fn like(&self, video_id: i32) {

}

fn dislike(&self, video_id: i32) {

}

fn get_likes_and_dislikes(&self, video_id: i32) -> Vec<i32> {

}

fn get_views(&self, video_id: i32) -> i32 {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* let obj = VideoSharingPlatform::new();
```

```
* let ret_1: i32 = obj.upload(video);
* obj.remove(videoId);
* let ret_3: String = obj.watch(videoId, startMinute, endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* let ret_6: Vec<i32> = obj.get_likes_and_dislikes(videoId);
* let ret_7: i32 = obj.get_views(videoId);
*/
```

Ruby:

```
class VideoSharingPlatform
def initialize()

end


=begin
:type video: String
:rtype: Integer
=end
def upload(video)

end


=begin
:type video_id: Integer
:rtype: Void
=end
def remove(video_id)

end


=begin
:type video_id: Integer
:type start_minute: Integer
:type end_minute: Integer
:rtype: String
=end
def watch(video_id, start_minute, end_minute)
```

```
end


=begin
:type video_id: Integer
:rtype: Void
=end
def like(video_id)

end


=begin
:type video_id: Integer
:rtype: Void
=end
def dislike(video_id)

end


=begin
:type video_id: Integer
:rtype: Integer[]
=end
def get_likes_and_dislikes(video_id)

end


=begin
:type video_id: Integer
:rtype: Integer
=end
def get_views(video_id)

end


end
```

```
# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform.new()
# param_1 = obj.upload(video)
# obj.remove(video_id)
# param_3 = obj.watch(video_id, start_minute, end_minute)
# obj.like(video_id)
# obj.dislike(video_id)
# param_6 = obj.get_likes_and_dislikes(video_id)
# param_7 = obj.get_views(video_id)
```

PHP:

```
class VideoSharingPlatform {
/**
*/
function __construct() {

}

/**
* @param String $video
* @return Integer
*/
function upload($video) {

}

/**
* @param Integer $videoId
* @return NULL
*/
function remove($videoId) {

}

/**
* @param Integer $videoId
* @param Integer $startMinute
* @param Integer $endMinute
* @return String
*/
function watch($videoId, $startMinute, $endMinute) {
```

```
}

/**
* @param Integer $videoId
* @return NULL
*/
function like($videoId) {

}

/**
* @param Integer $videoId
* @return NULL
*/
function dislike($videoId) {

}

/**
* @param Integer $videoId
* @return Integer[]
*/
function getLikesAndDislikes($videoId) {

}

/**
* @param Integer $videoId
* @return Integer
*/
function getViews($videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* $obj = VideoSharingPlatform();
* $ret_1 = $obj->upload($video);
* $obj->remove($videoId);
* $ret_3 = $obj->watch($videoId, $startMinute, $endMinute);
```

```
* $obj->like($videoId);
* $obj->dislike($videoId);
* $ret_6 = $obj->getLikesAndDislikes($videoId);
* $ret_7 = $obj->getViews($videoId);
*/
```

Dart:

```
class VideoSharingPlatform {

VideoSharingPlatform() {

}

int upload(String video) {

}

void remove(int videoId) {

}

String watch(int videoId, int startMinute, int endMinute) {

}

void like(int videoId) {

}

void dislike(int videoId) {

}

List<int> getLikesAndDislikes(int videoId) {

}

int getViews(int videoId) {

}
}
```

```
/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = VideoSharingPlatform();
* int param1 = obj.upload(video);
* obj.remove(videoId);
* String param3 = obj.watch(videoId,startMinute,endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* List<int> param6 = obj.getLikesAndDislikes(videoId);
* int param7 = obj.getViews(videoId);
*/
```

Scala:

```
class VideoSharingPlatform() {

def upload(video: String): Int = {

}

def remove(videoId: Int): Unit = {

}

def watch(videoId: Int, startMinute: Int, endMinute: Int): String = {

}

def like(videoId: Int): Unit = {

}

def dislike(videoId: Int): Unit = {

}

def getLikesAndDislikes(videoId: Int): Array[Int] = {

}

def getViews(videoId: Int): Int = {
```

```
}

}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* val obj = new VideoSharingPlatform()
* val param_1 = obj.upload(video)
* obj.remove(videoId)
* val param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* val param_6 = obj.getLikesAndDislikes(videoId)
* val param_7 = obj.getViews(videoId)
*/
```

Elixir:

```
defmodule VideoSharingPlatform do
@spec init_() :: any
def init_() do

end

@spec upload(video :: String.t) :: integer
def upload(video) do

end

@spec remove(video_id :: integer) :: any
def remove(video_id) do

end

@spec watch(video_id :: integer, start_minute :: integer, end_minute ::
integer) :: String.t
def watch(video_id, start_minute, end_minute) do

end

@spec like(video_id :: integer) :: any
```

```
def like(video_id) do

end

@spec dislike(video_id :: integer) :: any
def dislike(video_id) do

end

@spec get_likes_and_dislikes(video_id :: integer) :: [integer]
def get_likes_and_dislikes(video_id) do

end

@spec get_views(video_id :: integer) :: integer
def get_views(video_id) do

end
end

# Your functions will be called as such:
# VideoSharingPlatform.init_()
# param_1 = VideoSharingPlatform.upload(video)
# VideoSharingPlatform.remove(video_id)
# param_3 = VideoSharingPlatform.watch(video_id, start_minute, end_minute)
# VideoSharingPlatform.like(video_id)
# VideoSharingPlatform.dislike(video_id)
# param_6 = VideoSharingPlatform.get_likes_and_dislikes(video_id)
# param_7 = VideoSharingPlatform.get_views(video_id)

# VideoSharingPlatform.init_ will be called before every test case, in which
you can do some necessary initializations.
```

Erlang:

```
-spec video_sharing_platform_init_() -> any().
video_sharing_platform_init_() ->
.

-spec video_sharing_platform_upload(Video :: unicode:unicode_binary()) ->
integer().
video_sharing_platform_upload(Video) ->
```

```
.

-spec video_sharing_platform_remove(VideoId :: integer()) -> any().
video_sharing_platform_remove(VideoId) ->
.

-spec video_sharing_platform_watch(VideoId :: integer(), StartMinute ::
integer(), EndMinute :: integer()) -> unicode:unicode_binary().
video_sharing_platform_watch(VideoId, StartMinute, EndMinute) ->
.

-spec video_sharing_platform_like(VideoId :: integer()) -> any().
video_sharing_platform_like(VideoId) ->
.

-spec video_sharing_platform_dislike(VideoId :: integer()) -> any().
video_sharing_platform_dislike(VideoId) ->
.

-spec video_sharing_platform_get_likes_and_dislikes(VideoId :: integer()) ->
[integer()].
video_sharing_platform_get_likes_and_dislikes(VideoId) ->
.

-spec video_sharing_platform_get_views(VideoId :: integer()) -> integer().
video_sharing_platform_get_views(VideoId) ->
.


%% Your functions will be called as such:
%% video_sharing_platform_init_(),
%% Param_1 = video_sharing_platform_upload(Video),
%% video_sharing_platform_remove(VideoId),
%% Param_3 = video_sharing_platform_watch(VideoId, StartMinute, EndMinute),
%% video_sharing_platform_like(VideoId),
%% video_sharing_platform_dislike(VideoId),
%% Param_6 = video_sharing_platform_get_likes_and_dislikes(VideoId),
%% Param_7 = video_sharing_platform_get_views(VideoId),

%% video_sharing_platform_init_ will be called before every test case, in
which you can do some necessary initializations.
```

Racket:

```
(define video-sharing-platform%
(class object%
(super-new)

(init-field)

; upload : string? -> exact-integer?
(define/public (upload video)
)
; remove : exact-integer? -> void?
(define/public (remove video-id)
)
; watch : exact-integer? exact-integer? exact-integer? -> string?
(define/public (watch video-id start-minute end-minute)
)
; like : exact-integer? -> void?
(define/public (like video-id)
)
; dislike : exact-integer? -> void?
(define/public (dislike video-id)
)
; get-likes-and-dislikes : exact-integer? -> (listof exact-integer?)
(define/public (get-likes-and-dislikes video-id)
)
; get-views : exact-integer? -> exact-integer?
(define/public (get-views video-id)
)))

;; Your video-sharing-platform% object will be instantiated and called as
such:
;; (define obj (new video-sharing-platform%))
;; (define param_1 (send obj upload video))
;; (send obj remove video-id)
;; (define param_3 (send obj watch video-id start-minute end-minute))
;; (send obj like video-id)
;; (send obj dislike video-id)
;; (define param_6 (send obj get-likes-and-dislikes video-id))
;; (define param_7 (send obj get-views video-id))
```

Solutions

C++ Solution:

```
/*
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class VideoSharingPlatform {
public:
VideoSharingPlatform() {

}

int upload(string video) {

}

void remove(int videoId) {

}

string watch(int videoId, int startMinute, int endMinute) {

}

void like(int videoId) {

}

void dislike(int videoId) {

}

vector<int> getLikesAndDislikes(int videoId) {

}
```

```
int getViews(int videoId) {

}
};

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform* obj = new VideoSharingPlatform();
* int param_1 = obj->upload(video);
* obj->remove(videoId);
* string param_3 = obj->watch(videoId,startMinute,endMinute);
* obj->like(videoId);
* obj->dislike(videoId);
* vector<int> param_6 = obj->getLikesAndDislikes(videoId);
* int param_7 = obj->getViews(videoId);
*/
```

Java Solution:

```
/**
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class VideoSharingPlatform {

public VideoSharingPlatform() {

}

public int upload(String video) {

}

public void remove(int videoId) {
```

```
}

public String watch(int videoId, int startMinute, int endMinute) {

}

public void like(int videoId) {

}

public void dislike(int videoId) {

}

public int[] getLikesAndDislikes(int videoId) {

}

public int getViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = new VideoSharingPlatform();
* int param_1 = obj.upload(video);
* obj.remove(videoId);
* String param_3 = obj.watch(videoId,startMinute,endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* int[] param_6 = obj.getLikesAndDislikes(videoId);
* int param_7 = obj.getViews(videoId);
*/
```

Python3 Solution:

```
"""
Problem: Design Video Sharing Platform
Difficulty: Hard
```

```
Tags: array, string, tree, hash, stack

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

class VideoSharingPlatform:

def __init__(self):


def upload(self, video: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class VideoSharingPlatform(object):

def __init__(self):


def upload(self, video):
"""
:type video: str
:rtype: int
"""


def remove(self, videoId):
"""
:type videoId: int
:rtype: None
"""


def watch(self, videoId, startMinute, endMinute):
"""
:type videoId: int
:type startMinute: int
```

```
        :type endMinute: int
        :rtype: str
        """


    def like(self, videoId):
        """
        :type videoId: int
        :rtype: None
        """


    def dislike(self, videoId):
        """
        :type videoId: int
        :rtype: None
        """


    def getLikesAndDislikes(self, videoId):
        """
        :type videoId: int
        :rtype: List[int]
        """


    def getViews(self, videoId):
        """
        :type videoId: int
        :rtype: int
        """



# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform()
# param_1 = obj.upload(video)
# obj.remove(videoId)
# param_3 = obj.watch(videoId,startMinute,endMinute)
# obj.like(videoId)
# obj.dislike(videoId)
# param_6 = obj.getLikesAndDislikes(videoId)
```

```
# param_7 = obj.getViews(videoId)
```

JavaScript Solution:

```
/**
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/


var VideoSharingPlatform = function() {

};

/**
* @param {string} video
* @return {number}
*/
VideoSharingPlatform.prototype.upload = function(video) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.remove = function(videoId) {

};

/**
* @param {number} videoId
* @param {number} startMinute
* @param {number} endMinute
* @return {string}
*/
```

```
VideoSharingPlatform.prototype.watch = function(videoId, startMinute,
endMinute) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.like = function(videoId) {

};

/**
* @param {number} videoId
* @return {void}
*/
VideoSharingPlatform.prototype.dislike = function(videoId) {

};

/**
* @param {number} videoId
* @return {number[]}
*/
VideoSharingPlatform.prototype.getLikesAndDislikes = function(videoId) {

};

/**
* @param {number} videoId
* @return {number}
*/
VideoSharingPlatform.prototype.getViews = function(videoId) {

};

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = new VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
```

```
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

TypeScript Solution:

```
/**
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class VideoSharingPlatform {
constructor() {

}

upload(video: string): number {

}

remove(videoId: number): void {

}

watch(videoId: number, startMinute: number, endMinute: number): string {

}

like(videoId: number): void {

}

dislike(videoId: number): void {
```

```
}

getLikesAndDislikes(videoId: number): number[] {

}

getViews(videoId: number): number {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = new VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

C# Solution:

```
/*
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class VideoSharingPlatform {

public VideoSharingPlatform() {

}
```

```
public int Upload(string video) {

}

public void Remove(int videoId) {

}

public string Watch(int videoId, int startMinute, int endMinute) {

}

public void Like(int videoId) {

}

public void Dislike(int videoId) {

}

public int[] GetLikesAndDislikes(int videoId) {

}

public int GetViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = new VideoSharingPlatform();
* int param_1 = obj.Upload(video);
* obj.Remove(videoId);
* string param_3 = obj.Watch(videoId,startMinute,endMinute);
* obj.Like(videoId);
* obj.Dislike(videoId);
* int[] param_6 = obj.GetLikesAndDislikes(videoId);
* int param_7 = obj.GetViews(videoId);
*/
```

C Solution:

```
/*
* Problem: Design Video Sharing Platform
* Difficulty: Hard
* Tags: array, string, tree, hash, stack
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/




typedef struct {

} VideoSharingPlatform;


VideoSharingPlatform* videoSharingPlatformCreate() {

}

int videoSharingPlatformUpload(VideoSharingPlatform* obj, char* video) {

}

void videoSharingPlatformRemove(VideoSharingPlatform* obj, int videoId) {

}

char* videoSharingPlatformWatch(VideoSharingPlatform* obj, int videoId, int
startMinute, int endMinute) {

}

void videoSharingPlatformLike(VideoSharingPlatform* obj, int videoId) {

}

void videoSharingPlatformDislike(VideoSharingPlatform* obj, int videoId) {
```

```
}

int* videoSharingPlatformGetLikesAndDislikes(VideoSharingPlatform* obj, int
videoId, int* retSize) {

}

int videoSharingPlatformGetViews(VideoSharingPlatform* obj, int videoId) {

}

void videoSharingPlatformFree(VideoSharingPlatform* obj) {

}

/**
* Your VideoSharingPlatform struct will be instantiated and called as such:
* VideoSharingPlatform* obj = videoSharingPlatformCreate();
* int param_1 = videoSharingPlatformUpload(obj, video);

* videoSharingPlatformRemove(obj, videoId);

* char* param_3 = videoSharingPlatformWatch(obj, videoId, startMinute,
endMinute);

* videoSharingPlatformLike(obj, videoId);

* videoSharingPlatformDislike(obj, videoId);

* int* param_6 = videoSharingPlatformGetLikesAndDislikes(obj, videoId,
retSize);

* int param_7 = videoSharingPlatformGetViews(obj, videoId);

* videoSharingPlatformFree(obj);
*/
```

Go Solution:

```
// Problem: Design Video Sharing Platform
// Difficulty: Hard
// Tags: array, string, tree, hash, stack
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

type VideoSharingPlatform struct {

}


func Constructor() VideoSharingPlatform {

}


func (this *VideoSharingPlatform) Upload(video string) int {

}


func (this *VideoSharingPlatform) Remove(videoId int) {

}


func (this *VideoSharingPlatform) Watch(videoId int, startMinute int,
endMinute int) string {

}


func (this *VideoSharingPlatform) Like(videoId int) {

}


func (this *VideoSharingPlatform) Dislike(videoId int) {

}
```

```
func (this *VideoSharingPlatform) GetLikesAndDislikes(videoId int) []int {

}


func (this *VideoSharingPlatform) GetViews(videoId int) int {

}


/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* obj := Constructor();
* param_1 := obj.Upload(video);
* obj.Remove(videoId);
* param_3 := obj.Watch(videoId,startMinute,endMinute);
* obj.Like(videoId);
* obj.Dislike(videoId);
* param_6 := obj.GetLikesAndDislikes(videoId);
* param_7 := obj.GetViews(videoId);
*/
```

Kotlin Solution:

```
class VideoSharingPlatform() {

fun upload(video: String): Int {

}

fun remove(videoId: Int) {

}

fun watch(videoId: Int, startMinute: Int, endMinute: Int): String {

}

fun like(videoId: Int) {
```

```
}

fun dislike(videoId: Int) {

}

fun getLikesAndDislikes(videoId: Int): IntArray {

}

fun getViews(videoId: Int): Int {

}

}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* var obj = VideoSharingPlatform()
* var param_1 = obj.upload(video)
* obj.remove(videoId)
* var param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* var param_6 = obj.getLikesAndDislikes(videoId)
* var param_7 = obj.getViews(videoId)
*/
```

Swift Solution:

```
class VideoSharingPlatform {

init() {

}

func upload(_ video: String) -> Int {

}
```

```
func remove(_ videoId: Int) {

}

func watch(_ videoId: Int, _ startMinute: Int, _ endMinute: Int) -> String {

}

func like(_ videoId: Int) {

}

func dislike(_ videoId: Int) {

}

func getLikesAndDislikes(_ videoId: Int) -> [Int] {

}

func getViews(_ videoId: Int) -> Int {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* let obj = VideoSharingPlatform()
* let ret_1: Int = obj.upload(video)
* obj.remove(videoId)
* let ret_3: String = obj.watch(videoId, startMinute, endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* let ret_6: [Int] = obj.getLikesAndDislikes(videoId)
* let ret_7: Int = obj.getViews(videoId)
*/
```

Rust Solution:

```
// Problem: Design Video Sharing Platform
// Difficulty: Hard
```

```rust
// Tags: array, string, tree, hash, stack
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

struct VideoSharingPlatform {

}


/**
* `&self` means the method takes an immutable reference.
* If you need a mutable reference, change it to `&mut self` instead.
*/
impl VideoSharingPlatform {

fn new() -> Self {

}

fn upload(&self, video: String) -> i32 {

}

fn remove(&self, video_id: i32) {

}

fn watch(&self, video_id: i32, start_minute: i32, end_minute: i32) -> String
{

}

fn like(&self, video_id: i32) {

}

fn dislike(&self, video_id: i32) {

}
```

```
fn get_likes_and_dislikes(&self, video_id: i32) -> Vec<i32> {

}

fn get_views(&self, video_id: i32) -> i32 {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* let obj = VideoSharingPlatform::new();
* let ret_1: i32 = obj.upload(video);
* obj.remove(videoId);
* let ret_3: String = obj.watch(videoId, startMinute, endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* let ret_6: Vec<i32> = obj.get_likes_and_dislikes(videoId);
* let ret_7: i32 = obj.get_views(videoId);
*/
```

Ruby Solution:

```
class VideoSharingPlatform
def initialize()

end


=begin
:type video: String
:rtype: Integer
=end
def upload(video)

end


=begin
:type video_id: Integer
:rtype: Void
```

```
=end
def remove(video_id)

end


=begin
:type video_id: Integer
:type start_minute: Integer
:type end_minute: Integer
:rtype: String
=end
def watch(video_id, start_minute, end_minute)

end


=begin
:type video_id: Integer
:rtype: Void
=end
def like(video_id)

end


=begin
:type video_id: Integer
:rtype: Void
=end
def dislike(video_id)

end


=begin
:type video_id: Integer
:rtype: Integer[]
=end
def get_likes_and_dislikes(video_id)

end
```

```
=begin
:type video_id: Integer
:rtype: Integer
=end
def get_views(video_id)

end



end

# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform.new()
# param_1 = obj.upload(video)
# obj.remove(video_id)
# param_3 = obj.watch(video_id, start_minute, end_minute)
# obj.like(video_id)
# obj.dislike(video_id)
# param_6 = obj.get_likes_and_dislikes(video_id)
# param_7 = obj.get_views(video_id)
```

PHP Solution:

```
class VideoSharingPlatform {
/**
*/
function __construct() {

}

/**
* @param String $video
* @return Integer
*/
function upload($video) {

}

/**
```

```
* @param Integer $videoId
* @return NULL
*/
function remove($videoId) {

}

/**
* @param Integer $videoId
* @param Integer $startMinute
* @param Integer $endMinute
* @return String
*/
function watch($videoId, $startMinute, $endMinute) {

}

/**
* @param Integer $videoId
* @return NULL
*/
function like($videoId) {

}

/**
* @param Integer $videoId
* @return NULL
*/
function dislike($videoId) {

}

/**
* @param Integer $videoId
* @return Integer[]
*/
function getLikesAndDislikes($videoId) {

}

/**
```

```
* @param Integer $videoId
* @return Integer
*/
function getViews($videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* $obj = VideoSharingPlatform();
* $ret_1 = $obj->upload($video);
* $obj->remove($videoId);
* $ret_3 = $obj->watch($videoId, $startMinute, $endMinute);
* $obj->like($videoId);
* $obj->dislike($videoId);
* $ret_6 = $obj->getLikesAndDislikes($videoId);
* $ret_7 = $obj->getViews($videoId);
*/
```

Dart Solution:

```
class VideoSharingPlatform {

VideoSharingPlatform() {

}

int upload(String video) {

}

void remove(int videoId) {

}

String watch(int videoId, int startMinute, int endMinute) {

}

void like(int videoId) {
```

```
}

void dislike(int videoId) {

}

List<int> getLikesAndDislikes(int videoId) {

}

int getViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = VideoSharingPlatform();
* int param1 = obj.upload(video);
* obj.remove(videoId);
* String param3 = obj.watch(videoId,startMinute,endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* List<int> param6 = obj.getLikesAndDislikes(videoId);
* int param7 = obj.getViews(videoId);
*/
```

Scala Solution:

```
class VideoSharingPlatform() {

def upload(video: String): Int = {

}

def remove(videoId: Int): Unit = {

}

def watch(videoId: Int, startMinute: Int, endMinute: Int): String = {
```

```
}

def like(videoId: Int): Unit = {

}

def dislike(videoId: Int): Unit = {

}

def getLikesAndDislikes(videoId: Int): Array[Int] = {

}

def getViews(videoId: Int): Int = {

}

}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* val obj = new VideoSharingPlatform()
* val param_1 = obj.upload(video)
* obj.remove(videoId)
* val param_3 = obj.watch(videoId,startMinute,endMinute)
* obj.like(videoId)
* obj.dislike(videoId)
* val param_6 = obj.getLikesAndDislikes(videoId)
* val param_7 = obj.getViews(videoId)
*/
```

Elixir Solution:

```
defmodule VideoSharingPlatform do
@spec init_() :: any
def init_() do

end
```

```
@spec upload(video :: String.t) :: integer
def upload(video) do

end

@spec remove(video_id :: integer) :: any
def remove(video_id) do

end

@spec watch(video_id :: integer, start_minute :: integer, end_minute ::
integer) :: String.t
def watch(video_id, start_minute, end_minute) do

end

@spec like(video_id :: integer) :: any
def like(video_id) do

end

@spec dislike(video_id :: integer) :: any
def dislike(video_id) do

end

@spec get_likes_and_dislikes(video_id :: integer) :: [integer]
def get_likes_and_dislikes(video_id) do

end

@spec get_views(video_id :: integer) :: integer
def get_views(video_id) do

end
end

# Your functions will be called as such:
# VideoSharingPlatform.init_()
# param_1 = VideoSharingPlatform.upload(video)
# VideoSharingPlatform.remove(video_id)
# param_3 = VideoSharingPlatform.watch(video_id, start_minute, end_minute)
```

```
# VideoSharingPlatform.like(video_id)
# VideoSharingPlatform.dislike(video_id)
# param_6 = VideoSharingPlatform.get_likes_and_dislikes(video_id)
# param_7 = VideoSharingPlatform.get_views(video_id)

# VideoSharingPlatform.init_ will be called before every test case, in which you can do some necessary initializations.
```

Erlang Solution:

```
-spec video_sharing_platform_init_() -> any().
video_sharing_platform_init_() ->
.

-spec video_sharing_platform_upload(Video :: unicode:unicode_binary()) -> integer().
video_sharing_platform_upload(Video) ->
.

-spec video_sharing_platform_remove(VideoId :: integer()) -> any().
video_sharing_platform_remove(VideoId) ->
.

-spec video_sharing_platform_watch(VideoId :: integer(), StartMinute :: integer(), EndMinute :: integer()) -> unicode:unicode_binary().
video_sharing_platform_watch(VideoId, StartMinute, EndMinute) ->
.

-spec video_sharing_platform_like(VideoId :: integer()) -> any().
video_sharing_platform_like(VideoId) ->
.

-spec video_sharing_platform_dislike(VideoId :: integer()) -> any().
video_sharing_platform_dislike(VideoId) ->
.

-spec video_sharing_platform_get_likes_and_dislikes(VideoId :: integer()) -> [integer()].
video_sharing_platform_get_likes_and_dislikes(VideoId) ->
.
```

```
-spec video_sharing_platform_get_views(VideoId :: integer()) -> integer().
video_sharing_platform_get_views(VideoId) ->
.


%% Your functions will be called as such:
%% video_sharing_platform_init_(),
%% Param_1 = video_sharing_platform_upload(Video),
%% video_sharing_platform_remove(VideoId),
%% Param_3 = video_sharing_platform_watch(VideoId, StartMinute, EndMinute),
%% video_sharing_platform_like(VideoId),
%% video_sharing_platform_dislike(VideoId),
%% Param_6 = video_sharing_platform_get_likes_and_dislikes(VideoId),
%% Param_7 = video_sharing_platform_get_views(VideoId),

%% video_sharing_platform_init_ will be called before every test case, in
which you can do some necessary initializations.
```

Racket Solution:

```
(define video-sharing-platform%
(class object%
(super-new)

(init-field)

; upload : string? -> exact-integer?
(define/public (upload video)
)
; remove : exact-integer? -> void?
(define/public (remove video-id)
)
; watch : exact-integer? exact-integer? exact-integer? -> string?
(define/public (watch video-id start-minute end-minute)
)
; like : exact-integer? -> void?
(define/public (like video-id)
)
; dislike : exact-integer? -> void?
(define/public (dislike video-id)
)
```

```
; get-likes-and-dislikes : exact-integer? -> (listof exact-integer?)
(define/public (get-likes-and-dislikes video-id)
)
; get-views : exact-integer? -> exact-integer?
(define/public (get-views video-id)
)))

;; Your video-sharing-platform% object will be instantiated and called as
such:
;; (define obj (new video-sharing-platform%))
;; (define param_1 (send obj upload video))
;; (send obj remove video-id)
;; (define param_3 (send obj watch video-id start-minute end-minute))
;; (send obj like video-id)
;; (send obj dislike video-id)
;; (define param_6 (send obj get-likes-and-dislikes video-id))
;; (define param_7 (send obj get-views video-id))
```