

```

using System;
using System.Diagnostics;
using System.Collections.Generic;

class MySongList
{
    private int id;
    private string title;
    private int playCount;

    public MySongList(String title)
    {
        Debug.Assert(!string.IsNullOrEmpty(title), "Judul tidak boleh null atau kosong");
        if (string.IsNullOrEmpty(title)) throw new ArgumentException("Judul tidak boleh null atau kosong");

        Debug.Assert(title.Length < 200, "Judul tidak boleh lebih dari 200 karakter");
        if (title.Length > 200) throw new ArgumentException("Judul tidak boleh lebih dari 200 karakter");

        Random random = new Random();
        this.id = random.Next(10000, 99999);
        this.title = title;
        this.playCount = 0;
    }
}

```

Buat class MySongList dengan atribut id, title, dan playcount dengan title tidak boleh null atau maksimal 200 karakter menggunakan debug assert dan try catch, id random 4 digit, dan default play count = 0.

```

    public void IncreasePlayCount(int count)
    {
        Debug.Assert(count > 0 && count <= 25000000, "Count harus lebih dari 0 dan maksimal 25.000.000");
        if (count < 0 || count > 25000000) throw new ArgumentException("Count harus lebih dari 0 dan maksimal 25.000.000");

        try
        {
            checked
            {
                playCount += count;
            }
        }
        catch (OverflowException)
        {
            Console.WriteLine("⚠ Overflow terjadi! Play count tidak dapat ditambahkan.");
        }
    }

    public void PrintSongDetail()
    {
        Console.WriteLine($"ID: {id}, Title: {title}, Play Count {playCount}");
    }

    public String GetTitle() => title;

    public int GetPlayCount() => playCount;
}

```

Fungsi IncreasePlayCount dengan count tidak boleh negative dan maksimal 25.000.000 menggunakan debug assert dan try catch, kemudian menggunakan checked untuk mengetes overflow count. Fungsi print video detail menampilkan detail dari video dengan menyertakan atribut video.

```

class MySongUser
{
    private int id;
    private string username;
    private List<MySongList> uploadedSongs;

    public MySongUser(string username)
    {
        Debug.Assert(!string.IsNullOrEmpty(username), "Username tidak boleh null atau kosong");
        if (string.IsNullOrEmpty(username)) throw new ArgumentException("Username tidak boleh null atau kosong");

        Debug.Assert(username.Length <= 100, "Username tidak boleh lebih dari 100 karakter");
        if (username.Length > 100) throw new ArgumentException("Username tidak boleh lebih dari 100 karakter");

        Random random = new Random();
        this.id = random.Next(10000, 99999);
        this.username = username;
        this.uploadedSongs = new List<MySongList>();
    }
}

```

Class SayaTubeUser dengan atribut id, username, dan list dari class SayaTubeVideo dengan username tidak boleh null atau maksimal 100 karakter, id random 4 digit, kemudian list yang masih kosong.

```

public void AddSong(MySongList song)
{
    if (song == null) throw new ArgumentException("Song tidak boleh null");

    if (song.GetPlayCount() >= int.MaxValue)
    {
        throw new ArgumentException("Play count tidak boleh mencapai batas maksimum integer.");
    }

    uploadedSongs.Add(song);
}

```

Fungsi menambahkan video ke list dengan video tidak boleh null dan play count dari video yang ditambahkan tidak boleh lebih dari bilangan maksimal integer.

```

public int TotalPlayCount()
{
    int totalPlayCount = 0;
    foreach (var song in uploadedSongs)
    {
        totalPlayCount += 1;
    }
    return totalPlayCount;
}

public void PrintAllSongPlayCount()
{
    Console.WriteLine($"User: {username}");
    for (int i = 0; i < 8; i++)
    {
        Console.WriteLine($"Lagu {i + 1}: {uploadedSongs[i].GetTitle()}");
    }
}

```

Fungsi TotalPlayCount untuk mengembalikan berapa banyak video yang telah diputar oleh user. Kemudian fungsi PrintAllVideoPlaycount mengembalikan video yang diputar user dengan maksimal 8 video yang di kembalikan