1. Sự khác nhau giữa stack và heap?

Stack và heap là các vùng bộ nhớ. Heap nằm giữa stack và Permanent Storage area, và độ lớn có thể thay đổi khi thực thi chương trình.

1. Mục đích của thừa kế (inheritance)?

Để tái sử dụng.

1. Mục đích của đa hình (poymophism)?

Một hành động có thể xảy ra theo nhiều cách khác nhau.

1. Giảm dung lượng file APK

* Theo đó mấu chốt là theo dõi file Editor Log bằng cách nhấn chuột vào nút tam giác nhỏ ở góc trên bên phải cửa sổ Console và chọn <Open Editor Log>  
  rồi tìm đến đoạn có format tương tự như  
  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  Textures 3.8 mb 41.2%   
  Meshes 0.0 kb 0.0%   
  Animations 0.0 kb 0.0%   
  Sounds 18.3 kb 0.2%   
  Shaders 32.3 kb 0.3%   
  Other Assets 370.1 kb 3.9%   
  Levels 265.5 kb 2.8%   
  Scripts 636.6 kb 6.8%   
  Included DLLs 4.1 mb 44.1%
* Thông thường game nhỏ thì cần chú ý Textures và included DLLs
  + Với texture thì những file không dùng trong scene thì sẽ không được include trong file build. Nên việc xóa những file texture không cần thiết cũng không ảnh hưởng gì
  + Việc giảm dung lượng ban đầu của file ảnh cũng không ảnh hưởng gì. Do unity import và sử dụng với format riêng

1. Bài tập về mảng, list

<http://expressmagazine.net/development/460/bai-tap-5-tap-c-bai-tap-tong-hop-ve-mang>

Find second highest number in Array

int[] myArray = new int[] {1, 3, 2, 5, 4, 8, 20, 99};  
  
void Start()  
{  
 Debug.Log(GetSecondHighest());  
 Debug.Log("min " + GetMin());  
 Debug.Log("Second lowest " + GetSecondLowest());  
}  
  
int GetSecondHighest()  
{  
 return myArray.OrderByDescending(x => x).Distinct().Skip(1).First();  
}  
  
int GetMin()  
{  
 return myArray.Min();  
}  
  
int GetSecondLowest()  
{  
 return myArray.OrderBy(x => x).Distinct().Skip(1).First();  
}

List

List<int> myList = new List<int>() {1, 3, 5, 7, 2, 4};

int GetList()  
{  
 myList.Sort();  
 return (myList[myList.Count-2]);  
}

Chẵn lẻ trong mảng

int CountOdd() // odd : so le  
{  
 foreach (var i in oddEvenArray)  
 {  
 if (i % 2 != 0)  
 oddCount++;  
 }  
 return oddCount;  
}