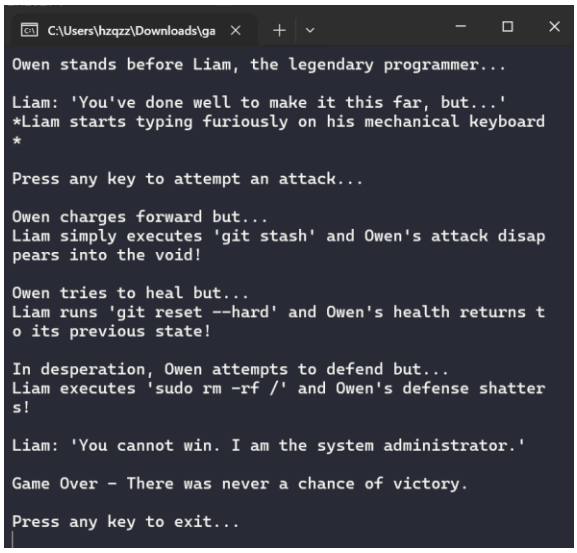
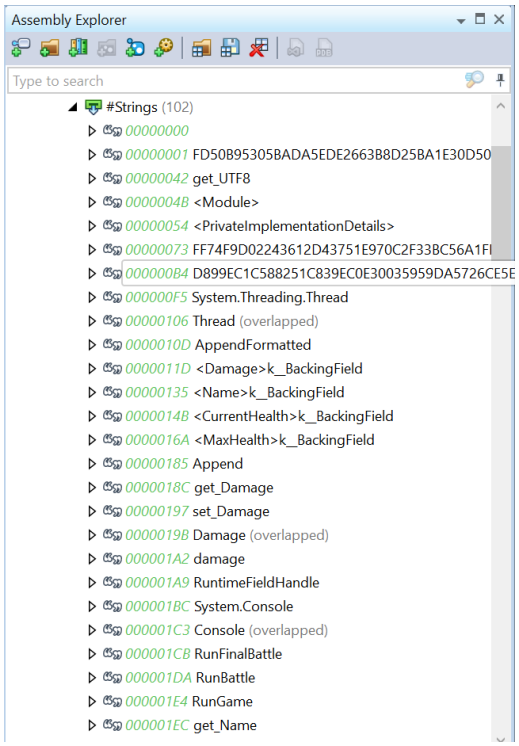
Reverse Engineering -Bossman

I use Jetbrain dotpeek. Using Dnspy cannot see the source code. But dotpeek cant modify code or im too noob. Run the file to see which strings appear.



Then locate the strings in dotpeek to locate the main function



```
// Decompiled with JetBrains decompiler
// Type: Program
// Assembly: source.cs, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// MVID: B6A64903-B068-4606-93DF-F54E7FA5A49C
// Assembly location: source.cs.dll inside C:\Users\hzqzz\Downloads\game.bossman.stout.exe)

using System;
using System.Diagnostics;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading;

#nullable enable
internal class Program
{
  private static void Main(string[] args)
  {
    Console.WriteLine("Welcome to the Mini STOUT MMO Adventure!");
    Console.WriteLine("\nIn a world of CCDC competitions, our hero Owen embarks on a perilous journey...");
    Console.WriteLine("Press any key to begin the adventure...");
    Console.ReadKey();
    Console.Clear();
    Program.RunGame(new Program.Character("Owen", 50, 10));
  }

  private static void RunGame(Program.Character hero)
  {
    Console.WriteLine("\n=== Room 1: The CNIT Lab ===");
    Console.WriteLine("Owen enters the dimly lit CNIT lab. The hum of computers fills the air...");
    Console.WriteLine("Suddenly, Holly appears, wielding a mechanical keyboard controlling a powerpoint!");
    Console.WriteLine("\nPress any key to begin the battle...");
    Console.ReadKey();
    Program.Character boss1 = new Program.Character("Holly", 20, 2);
    if (!Program.RunBattle(hero, boss1))
      return;
    Console.WriteLine("\n=== Room 2: The JSP Room ===");
    Console.WriteLine("Owen pushes forward into the JSP room, where ancient servlets still run...");
```

Looking into the code we can see that the function getTime() is the only that have some sort of encoding process.

```
private static string getTime()
{
  char[] chArray1 = new char[4]{ 'U', '1', 'R', 'P' };
  char[] chArray2 = new char[4]{ 'V', 'V', 'R', 'D' };
  char[] chArray3 = new char[4]{ 'V', 'E', 'Z', '7' };
  char[] charArray1 = "VGZSVDBsWW9nUjRWa3ZDaEdab2tS".ToCharArray();
  char[] charArray2 = "MkFmR2ZUQ0p2SEh9".ToCharArray();
  StringBuilder stringBuilder = new StringBuilder();
  foreach (char ch in chArray1)
    stringBuilder.Append((char) ((uint) ch ^ 0U));
  for (int index = 0; index < chArray2.Length; ++index)
    stringBuilder.Append(chArray2[index].ToString().ToCharArray()[0]);
  stringBuilder.Append(new string(chArray3));
  stringBuilder.Append(Convert.ToBase64String(Encoding.UTF8.GetBytes(new string(charArray1))).Substring(0, charArray1.Length));
  stringBuilder.Append(Encoding.UTF8.GetString(Encoding.UTF8.GetBytes(new string(charArray2))));
  return stringBuilder.ToString();
}

private static void DefeatLiam()
{
  Console.WriteLine("#git blame the one who wrote this " + Program.getTime());
}
```

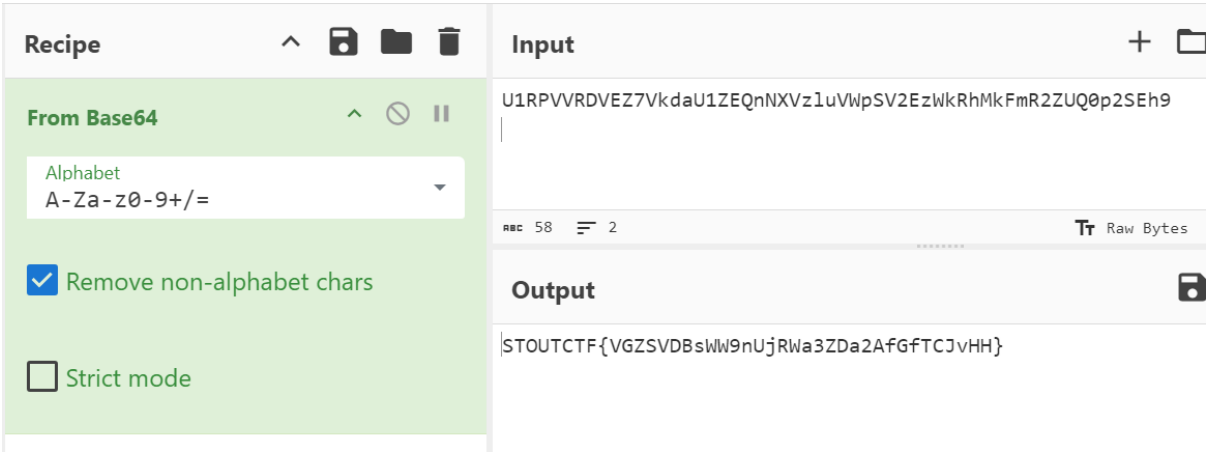
Another ChatGPT. Change the Java to python so I can run it easier

```
import base64

# Data from the decompiled getTime() function
chArray1 = ['U', '1', 'R', 'P']
chArray2 = ['V', 'V', 'R', 'D']
chArray3 = ['V', 'E', 'Z', '7']
charArray1 = "VGZSVDBsWW9nUjRWa3ZDaEdab2tS"
charArray2 = "MkFmR2ZUQ0p2SEh9"

# Initialize the result list
result = []

# Add chArray1 to result (XORed with 0 which does nothing here)
```

```
result.extend(chArray1)

# Add chArray2 to result
result.extend(chArray2)

# Add chArray3 to result
result.extend(chArray3)

# Encode charArray1 to Base64 and slice it to its original length
base64_encoded_charArray1 = base64.b64encode(charArray1.encode('utf-8')).decode('utf-8')
result.append(base64_encoded_charArray1[:len(charArray1)])

# Append charArray2 (interpreted as a UTF-8 string)
result.append(charArray2)

# Join the result to form the final string
final_string = ''.join(result)

# Print the final string
print(final_string)
```

Just run it and got base64. Why don't I decode base64 in the code? I did not think of it

U1RPVVRDVEZ7VkdaU1ZEQnNXVzluVWpSV2EzWkRhMkFmR2ZUQ0p2SEh9



STOUTCTF{VGZSVDBsWW9nUjRWa3ZDa2AfGfTCJvHH}