Ozyegin University
CS 321 Programming Languages
Sample Problems on Garbage Collection

1. Which garbage collection algorithm would work the best for the following piece of Java program? Explain why. Remember that Strings in Java are immutable objects (i.e. once created, you may not modify them).

```java
String concat(String[] ss) {
  // Assume ss is very long
  String res = "";
  for (int i = 0; i < ss.length; i++)
      res += ss[i];
  return res;
}
```

   Suggest a better (i.e. more efficient) way to implement this code. Explain why it would be more efficient.

2. Write a piece of program (in Java or C++) that would cause a memory leak if reference counting were used as the garbage collection technique.

3.

(a) Suppose I have a program where I create a long linked list structure with no cycles, and then set my pointer to the head of the list to null, so the head becomes unreachable. Which garbage collection algorithm would perform the best for this case? Justify by explaining what drawbacks the other algorithms have for this case.

(b) Which garbage collection algorithm is likely to improve cache utilization? Why?

(c) Suppose I have a computer with small heap memory. If I am to make a choice between mark-and-sweep and two-space-stop-and-copy algorithms, which one should I choose? Why?

(d) Suppose I have a program where I create a lot of temporary (i.e. short-lived) objects of various sizes scattered (i.e. *saçılmış, yayılmış*) all over the memory. If I am to make a choice between mark-and-sweep and two-space-stop-and-copy algorithms, which one should I choose? Why?

(e) Suppose I have a program where I create many long-lived and large objects and arrays. If I am to make a choice between mark-and-sweep and two-space-stop-and-copy algorithms, which one should I choose? Why?