

# CSCI 135 Software Design and Analysis, C++

## Lab 9

### Solution

Saad Mneimneh  
Hunter College of CUNY

#### Lab A: Split

Write a function called `split` that has the following signature:

```
char * split(const char * s, char * t, bool b) {  
    .  
    .  
    .  
    return t;  
}
```

If `b` is true, it copies all even numbered characters in `s` to `t`. Otherwise, it copies all odd numbered characters in `s` to `t`. For example, if `s` is “hello”, `split(s,t,true)` will make `t` “hlo”, and `split(s,t,false)` will make `t` “el”. Assume `t` is long enough and don’t forget to append the null character in both cases.

#### Lab B: Merge

Write a function called `merge` that has the following signature:

```
char * merge(const char * s1, const char * s2, char *t) {  
    .  
    .  
    .  
    return t;  
}
```

This function merges `s1` and `s2` into `t` starting with `s1`. For example, if `s1` is “hello” and `s2` is “abcdefg”, then `t` will be “haeblcldoefg”. Assume `t` is long enough and don’t forget the null character. Also, make sure that

```
merge(split(s,t1,true), split(s,t2,false), t);
```

makes `t` a copy of `s`.

## Solution:

```
#include <iostream>
#include <cstring> //standard c strings
using std::cout;

char * split(const char * s, char * t, bool b) {
    int i;
    int l=strlen(s);
    //I will interpret b as 0 or 1
    int offset=1-b; //offset is either 0 or 1
    for (i=0; 2*i+offset<l; i=i+1) {
        t[i]=s[2*i+offset];
    }
    t[i]='\0';
    return t;
}

int min(int a, int b) {
    if (a<=b)
        return a;
    else
        return b;
}

char * merge(const char * s1, const char * s2, char * t) {
    int l1=strlen(s1);
    int l2=strlen(s2);
    int l=min(l1,l2);
    //copy the first l characters of each
    for (i=0; i<l; i=i+1) {
        t[2*i]=s1[i];
        t[2*i+1]=s2[i];
    }
    //copy the rest starting at appropriate positions
    if (l1>=l2)
        strcpy(t+2*i, s1+l);
    else
        strcpy(t+2*i, s2+l);
    return t;
}

int main() {
    char s[]='hellosaad';
    char t1[10];
    char t2[10];
    char t[10];
    merge(split(s,t1,true), split(s,t2,false), t);
    cout<<t1<<'\n';
    cout<<t2<<'\n';
    cout<<t<<'\n';
}
```

