

# Checking for Palindromes Done Recursively

Matt Bishop

ECS 36A, Fall Quarter 2023

UC Davis

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

```
14:
15: int main(void)
16: {
17:     char buf[1000];
18:     (void) strcpy(buf, "madam")
19:     if (ispal(buf))
20:         printf("Palindrome\n");
21:     else
22:         printf("Not a palindrome\n");
23:     return(0);
24: }
```

Initial call to ispal: ispal(n ← “madam”)

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]){
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

ispal(“madam”): return to main, line 19  
n = “madam”

ispal(n ← "madam"):

4: condition false, so skip

9: call ispal("ada")

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

ispal("ada"): return to line 10, purple arrow

n = "ada"

ispal("madam"): return to main, line 19  
n = "madam"

ispal(n ← "ada"):

6: condition false, so skip

9: call ispal("d")

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

ispal("d"): return to line 10, red arrow  
n = "d"

ispal("ada"): return to line 10, purple  
arrow  
n = "ada"

ispal("madam"): return to main, line 19  
n = "madam"

ispal(n ← "d"):  
6: condition true, so return 1

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:         ↑
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

ispal("d"): do not recurse, blue arrow

ispal("d"): return to line 10, red arrow  
n = "d"

ispal("ada"): return to line 10, purple arrow  
n = "ada"

ispal("madam"): return to main, line 19  
n = "madam"

ispal(n ← "d"):  
6: condition true, so return 1

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:         ↑
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

~~ispal(""): line 5, returns 1~~

ispal("d"): return to line 10, red arrow  
n = "d"

ispal("ada"): return to line 10, purple  
arrow  
n = "ada"

ispal("madam"): return to main, line 19  
n = "madam"

ispal(n ← "d"):  
at line 10, return 1

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

~~ispal("d"): at line 10, ispal is 1, so return 1~~

ispal("ada"): return to line 10, purple arrow


n = "ada"

ispal("madam"): return to main, line 19  
n = "madam"



ispal(n ← "ada"):  
at line 10, return 1

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```



~~ispal("ada"): at line 10, ispal returns 1,  
so this returns 1 also~~

ispal("madam"): return to main, line 19  
n = "madam"

ispal(n ← "madam"):  
at line 10, return 1

```
1: int ispal(char *n)
2: {
3:     /* base case */
4:     if (!*n || strlen(n) == 1)
5:         return(1);
6:
7:     /* recurse case */
8:     if (*n == n[strlen(n)-1]) {
9:         n[strlen(n)-1] = '\\0';
10:        return(ispal(n+1));
11:    }
12:    else return(0);
13: }
```

~~ispal("madam"): at line 10, ispal returns  
1, so this also returns 1~~