

## Swap, Case Insensitive Search

# CS 241

## Data Organization using C

Instructor: **Joel Castellanos**

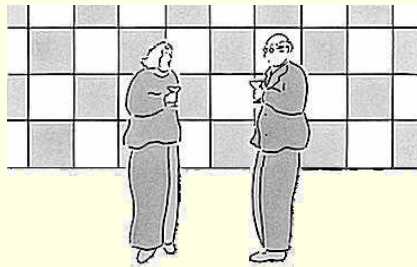
e-mail: [joel@unm.edu](mailto:joel@unm.edu)

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering  
Center (FEC) room 321

Lab Instructor: **Dongye Meng**

e-mail: [dymeng@cs.unm.edu](mailto:dymeng@cs.unm.edu)



3/24/2009

*"I am case sensitive. So, please, barbara with a small b."*

## Swap

```
void swap(int *a, int *b)
{ int tmp = *a;
  *a = *b;
  *b = tmp;
}
main()
{ int x,y;
  x = 9;
  y = 7;
  printf("x=%d, and y=%d\n",x,y);
  swap(&x,&y);
  printf("x=%d, and y=%d\n",x,y);
}
```

Output:

```
x=9, and y=7
x=7, and y=9
```

## charCmpCaseInsensitive()

```
int charCmpCaseInsensitive(char c1, char c2)
{ int lowerCaseOffset = 'A' - 'a';
  if (c1 >= 'a' && c1 <= 'z')
  { c1 += lowerCaseOffset;
  }
  if (c2 >= 'a' && c2 <= 'z')
  { c2 += lowerCaseOffset;
  }
  return c1==c2;
}
```

3

## findSubstringCaseInsensitive()

```
char *findSubstringCaseInsensitive(
    char *haystack, char *needle)
{ int len = strlen(needle);
  int matchCount = 0;
  while (*haystack)
  { if ( charCmpCaseInsensitive(
        *(needle+matchCount), *haystack)
      { matchCount++;
        if (matchCount == len)
        { char *startPt = (haystack - len)+1;
          return startPt;
        }
      }
    else {haystack -= matchCount; matchCount = 0;}
    haystack++;
  }
  return NULL;
}
```

4

## Quiz: Substring Search

```
char *findSubstring(char *str, char *needle)
{ int len = strlen(needle);
  int n = 0;
  while (*str)
  { printf("%c%c ", *str, *needle);
    if ( *(needle+n) == *str)
    { n++;
      if (n == len) return (str-len)+1;
    }
    else
    { str -= n;
      n = 0;
    }
    str++;
  }
  return NULL;
}
```

What is the output of:

findSubstring("ABCDE", "CD")

- a) AC BC CC DD
- b) AC BC CC DC
- c) AC BC CC DC EC
- d) AC BC CC DC ED
- e) AC BC CC

5

## Quiz: Substring Search

```
char *findSubstring(char *str, char *needle)
{ int len = strlen(needle);
  int n = 0;
  while (*str)
  { printf("%c%c ", *str, *(needle+n));
    if ( *(needle+n) == *str)
    { n++;
      if (n == len) return (str-len)+1;
    }
    else
    { str -= n;
      n = 0;
    }
    str++;
  }
  return NULL;
}
```

What is the output of:

findSubstring("ACDCDEF", "CDE")

- a) AC CC DC CC DC CC DC EC
- b) AC CC DC CC DC CC DD EE
- c) AC CC DD CE DE CE DE EE
- d) AC CC DD CE DC CC DD EC
- e) AC CC DD CE DC CC DD EE

6