APS105 Lecture 20
8.3

Dan Zingaro

March 12, 2010

Feedback from Reading Quiz

- "When you are trying to write a program using recursion, should you figure out what the base case is before you figure out the general case? Or is it better to identify the base case after finding the general case?"
  - Usually, the base case is easier to figure out
  - e.g. the number 0, the string with no characters ...

Reversing Numbers

- We will write a function `printRev(n)` that prints a positive integer in reverse
  - e.g. `printRev(1234)` prints `4321`
- To use recursion, we must be able to break the problem into simpler subproblems, and then be able to combine them to solve the original problem
- We can get the rightmost digit of `n` using `n % 10`
- We can get all digits of `n`, except the rightmost digit, using `n / 10`
- For a number like `1234`, we want to print the `4`, and then print the reverse of `123`
  - `1234 % 10` is `4` (the rightmost digit)
  - `1234 / 10` is `123` (all but the rightmost digit)

ConcepTest

Which of the following correctly prints a number in reverse?

- A.

```
void printRev1 (int n) {
  if (n < 10)
    printf ("%d", n);
  else {
    printf ("%d", n % 10);
    printRev1 (n / 10);
  }
}
```

- B.

```
void printRev2 (int n) {
  if (n < 10)
    printf ("%d", n);
  else {
    printRev2 (n / 10);
    printf ("%d", n % 10);
  }
}
```

- C. A and B are correct

Strings: Thinking Recursively

- We have been thinking of a string as a sequence of characters
- We can also think of the structure of a string recursively
  - 1. The `'\0'` character is a string
  - 2. A character followed by a string is a string
- Or
  - 1. The `'\0'` character is a string
  - 2. A string followed by a character is a string
- e.g. why is `"the"` a string according to these definitions?

Recursively Finding Length of a String

- Here is a recursive definition of the length of a string
  - 1. The length of the string `'\0'` is 0
  - 2. Otherwise, the length of a string is 1 plus the length of the string without its first character

ConcepTest

Which of the following correctly finds the length of string s?

- A.

```
int findLen1 (char *s) {
  if (*s == '\0')
    return 0;
  else
    return 1 + findLen1 (s+1);
}
```

- B.

```
int findLen2 (char *s) {
  if (*s == '\0')
    return 1;
  else
    return 1 + findLen2 (s+1);
}
```

- C.

```
int findLen3 (char *s) {
  if (*s == '\0')
    return 0;
  else
    return findLen3 (s+1) + 1;
}
```

- D. Two of the above are correct
- E. All of the above are correct

ConcepTest

Which of the following returns `true` if and only if `i` is a positive even integer?

- A.

```
bool isEven1 (int i) {
  if (i == 2) return true;
  else if (i == 1) return false;
  else
    return isEven1 (i + 2);
}
```

- B.

```
bool isEven2 (int i) {
  if (i == 2) return true;
  else if (i == 1) return false;
  else
    return isEven2 (i - 2);
}
```

- C.

```
bool isEven3 (int i) {
  if (i == 2) return true;
  else
    return isEven3 (i - 2);
}
```

ConcepTest

Which of the following returns the number of occurrences of character c in string s?

- A.

```
int num1 (char *s, char c) {
  if (*s == c)
    return 1 + num1 (s + 1, c);
  else
    return 0 + num1 (s + 1, c);
}
```

- B.

```
int num2 (char *s, char c) {
  if (*s == '\0')
    return 0;
  else if (*s == c)
    return 1 + num2 (s + 1, c);
  else
    return 0 + num2 (s + 1, c);
}
```

- C. A and B are correct