Pointer arithmetic and linked lists in C

1/27/25

Administrivia

- HW 3 (memory diagrams and binary) due tonight

- Before class on Wednesday, read Chapter 11 through 11.3 from Dive Into Systems (beginning of the Storage chapter)

Find first non-space

Suppose you have a string called line

Find its first non-space char (set i to its index)

Find first non-space

Suppose you have a string called line

Find its first non-space char (set i to its index)

```
int i = 0;                      //index into line
while(line[i] == ' ')           //advance i to 1st non-space
   i++;
```

Copying a substring

Suppose line stores a string and word is a char*

Copy the first word (up to space) starting at line[i] to word

Copying a substring

Suppose line stores a string and word is a char*

Copy the first word (up to space) starting at line[i] to word

```
int j = 0;                //index in word
while((line[i] != ‘ ‘) && (line[i] != 0)) {     //while in word
    word[j] = line[i];        //copy next char
    i++;                      //advance indices
    j++;
}
word[j] = 0;                        //add terminating 0
```

Yuck!

- Both use string[index] construct
  - potentially lots of indices
  - hard to print part of a word (requires a copy)
  - arguably, lots of typing

Alternative: Pointer arithmetic

- Use a pointer into the array
- Pointer itself moves: ptr++ advances it
  - Can also use other arithmetic
  - adding “1” moves address by 1 cell (**not** 1 byte)
- Access value at pointer’s location with *ptr

Finding first non-space revisited

Suppose you have a string called line

Find its first non-space char (set i to its index)

```
char* ptr = line;          //pointer into line
while(*ptr == ' ')         //advance i to 1st non-space
   ptr++;
```

If A is an array and it has been assigned to the pointer ptr, what does *ptr give?

A. A syntax error
B. Cell 0 of array A
C. Some other cell of array A
D. The size of array A
E. Depends on the type of array A

If A is an array and it has been assigned to the pointer ptr, what does *ptr give?

A. A syntax error
B. Cell 0 of array A
C. Some other cell of array A
D. The size of array A
E. Depends on the type of array A

If A is an array and it has been assigned to the pointer ptr, what does *(ptr+1) give?

A. A syntax error
B. Cell 0 of array A
C. Cell 1 of array A
D. The size of array A
E. Depends on the type of array A

If A is an array and it has been assigned to the pointer ptr, what does *(ptr+1) give?

A. A syntax error
B. Cell 0 of array A
C. Cell 1 of array A
D. The size of array A
E. Depends on the type of array A

If A is an array and it has been assigned to the pointer ptr, what does ptr++ do?

A. Give a syntax error
B. Increment the value stored in cell 0 of array A
C. Advance ptr so it stores the address one greater than the address of cell 0 of array A
D. Advance ptr so it stores the address of cell 1 of array A
E. Depends on the type of array A

If A is an array and it has been assigned to the pointer ptr, what does ptr++ do?

A. Give a syntax error
B. Increment the value stored in cell 0 of array A
C. Advance ptr so it stores the address one greater than the address of cell 0 of array A
D. Advance ptr so it stores the address of cell 1 of array A
E. Depends on the type of array A

What does the following code do?

```
void f(char* a, char* b) {
    while((*a++ = *b++));
}
```

A. Advances a and b to point to the first character at which they differ
B. Copies string b to string a
C. Increments characters in strings a and b
D. Memory error (seg fault/bus error)
E. Something unpredictable

What does the following code do?

```
void f(char* a, char* b) {
    while((*a++ = *b++));
}
```

A. Advances a and b to point to the first character at which they differ
B. Copies string b to string a
C. Increments characters in strings a and b
D. Memory error (seg fault/bus error)
E. Something unpredictable

What does the following code do?

```
int f(char* a) {
    char* b = a;
    while(*b++);
    return b-a-1;
}
```

A. Returns the length of string a
B. Changes the first char of string a to \0 and returns -1
C. Returns the first index whose character matches the string's first character (or the string length)
D. Memory error (seg fault/bus error)
E. Infinite loop

What does the following code do?

```
int f(char* a) {
    char* b = a;
    while(*b++);
    return b-a-1;
}
```

A. Returns the length of string a
B. Changes the first char of string a to \0 and returns -1
C. Returns the first index whose character matches the string's first character (or the string length)
D. Memory error (seg fault/bus error)
E. Infinite loop

What about this?

```
int g(char* a, char* b) {
    while(*a++ == *b++)
        if(*(a-1) == 0)
            return 0;
    return *(a-1) - *(b-1);
}
```