

Pointers to Pointers



Review: Arrays and Pointers

- A vector is a list of contiguous data items
 - e.g. “`int grades[10];`” is a list of 10 integers
- We can use the name of the vector as a pointer to the first element in that list
 - `grades == &grades[0]`
- A pointer is actually a pointer to zero, one, or more items
 - `int *gptra=grades;`
- Anything we can do with an array, we can also do with pointers...

Example Average Calculation

With Arrays

```
int grades[10]={97,98,82,...};
int j; int sum=0;
for(j=0;j<10;j++) {
    sum+=grades[j];
}
float avg=sum/10.0;
```

With Pointers

```
int grades[10]={97,98,82,...};
int *gptr; int sum=0;
for(gptr=grades;gptr<grades+10;gptr++) {
    sum+=(*gptr);
}
float avg=sum/10.0;
```

Q: How do we manage lists of lists?

- For example, a recipe has a list of ingredients
 - how do we handle a list of recipes?
- For example, each student has a list of grades
 - how do we handle a list of students?
- For example, a string is a list of characters
 - how do we handle a list of strings?
 - words in a sentence
 - Sentences in a paragraph
 - Paragraphs in a chapter
 - Chapters in a book

Answer 1: Matrixes (2D lists)

- For instance, in lab 6 (arrays) we had
 - `float grades[MAX_STUDENTS][MAX_GRADES];`
 - Each row represented the grades for one student
 - Each column represented the grades for one test
- This is a case where each row has the same number of columns
 - Each student took the same list of tests
- What happens when each row has a different number of columns?

Example: List of sentences

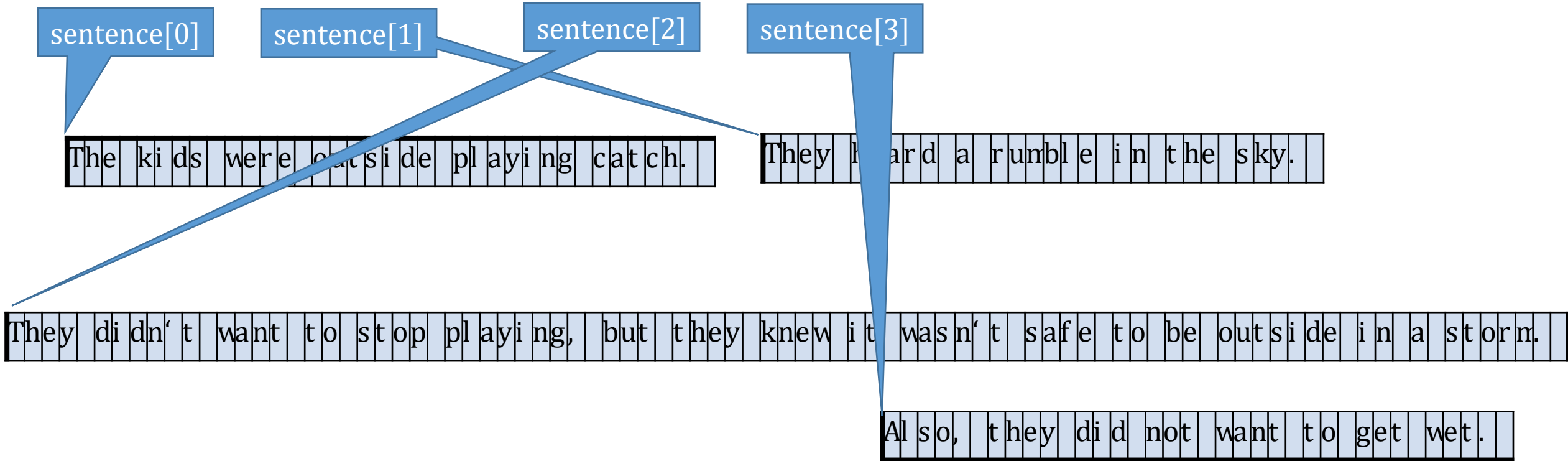
- The kids were outside playing catch.
- They heard a rumble in the sky.
- They didn't want to stop playing, but they knew it wasn't safe to be out in a storm.
- Also, they did not want to get wet.

Sentences in a Matrix

The	kids	were	outside	playing	catch.
They	heard	a	rumble	in	the sky.
They	didn't	want	to stop	playing,	but they knew it wasn't safe to be outside in a storm.
Also,	they	did	not want	to get wet.	

- Problem.... LOTS of wasted space!
- Every row must have the same number of columns!
- Every row must be as long as the longest row

Alternative... Lists of Pointers



Vector of Pointers in C

```
char * sentence[4] =  
    { "The kids were outside playing catch",  
      "They heard a rumble in the sky",  
      "They didn't want to stop playing, but they knew it wasn't  
safe to be out in a storm.",  
      "Also, they did not want to get wet"};
```

Using Pointer Notation

```
char ** sentence =  
    { "The kids were outside playing catch",  
      "They heard a rumble in the sky",  
      "They didn't want to stop playing, but they knew it wasn't  
safe to be out in a storm.",  
      "Also, they did not want to get wet"};
```

Reading **

- `char **sentence;`
 - “sentence is a pointer to a pointers to a character”
 - But, pointers are pointers to one or more... so
- “sentence is a pointer to one or more pointers to one or more characters”
 - But since a “pointer to one or more characters” is a string...
- “sentence is a pointer to one or more strings”
 - But since a pointer to one or more is a vector of
- “sentence is a vector of strings”

Understanding Run-Time Parameters

> ./mypgm this is a test

- Operating system parses this input string
 - One or more white-space characters divide parameters
 - OS includes the name of the program itself
 - OS creates an array of strings

Understanding Run-Time Parameters

> ./mypgm this is a test

- OS Creates....

```
int argc=5;
```

```
char **argv={"./mypgm","this","is","a","test"};
```

- ... Then OS calls main...

```
retcode=main(argc,argv);
```

Example Printing Run-Time Parameters

```
#include <stdio.h>
int main(int argc, char**argv) {
    int i;
    for(i=0;i<argc,i++) {
        printf("%2d: %s\n",i,argv[i]);
    }
    return 0;
}
```

```
>./mypgm this is a test
0: ./mypgm
1: this
2: is
3: a
4: test
>
```

Resources

- Programming in C, Chapter 7
- Wikipedia Pointers
[https://en.wikipedia.org/wiki/Pointer_\(computer_programming\)](https://en.wikipedia.org/wiki/Pointer_(computer_programming))
- Wikipedia Entry Point
[https://en.wikipedia.org/wiki/Entry_point#C and C.2B.2B](https://en.wikipedia.org/wiki/Entry_point#C_and_C.2B.2B)
- Command line arguments tutorial
<http://www.cprogramming.com/tutorial/c/lesson14.html>