

# CSE 251 - last one

Arend Hintze

# homework - linked list

- read a file once
- find all unique words
- count how often each unique word appears
- suggestion: use a linked list...

root ->

```
typedef struct node{  
    char *word;  
    int count;  
    struct node *next;  
} Node;
```

"one"

runner

```
typedef struct node{  
    char *word;  
    int count;  
    struct node *next;  
} Node;
```

"two"

dummy

```
typedef struct node{  
    char *word;  
    int count;  
    struct node *next;  
} Node;
```

"three"

NULL

# displaying the list

```
runner=root;
while(runner!=NULL){
    printf("%s %i\n",runner->word,runner->count);
    runner=runner->next;
}
```

**on the board**

# make your own include file

- we used `#include <something.h>`
- you can write your own include file
- use it by `#include "myFileName.h"`
- make it include it once only

```
#ifndef myFileName_h  
#define myFileName_h
```

```
#endif
```

```
#ifndef myFileName_h
#define myFileName_h
```

## prototypes:

```
void resetList(void);
void addCharToList(char theChar);
int getCountOfChar(char theChar);
bool hasChar(char theChar);
void addCountToChar(char theChar, int theCount);
char* getCharArray(void);
int* getIntArray(void);
int lengthOfList(void);
```

```
#endif
```



```
#ifndef myFileName_h
#define myFileName_h
```

## prototypes:

```
void resetList(void);
void addCharToList(char theChar);
int getCountOfChar(char theChar);
bool hasChar(char theChar);
void addCountToChar(char theChar, int theCount);
char* getCharArray(void);
int* getIntArray(void);
int lengthOfList(void);
```

## implementations:

```
void resetList(void){
    ...
}
```

```
#endif
```

# first exam topics

- basic input output
- variables
- loops (while, for)
- flow control (if, switch)
- functions
- arrays

# next exam topic

- basics knowledge required!
- multidimensional arrays
- dynamic arrays
- pointers ...
- file IO
- ...structures
- all of that combined!

# typical question regarding arrays

- you get a 3D array of int find the lowest, highest, closest to mean
- same array filled with 0-9s count how many 0-9 you find
- same array filled with arbitrary numbers count how often you find each <similar to dynamic arrays...>
- remember magic squares?

# functions and malloc

- make a function that returns an array of int from S to E with I increments
- take an array and check if it starts with S, goes to E, and has I increments
- make a function that returns a 3D array filled with random numbers (or other weird constraint)
- you get two arrays: one is an array of pointers to int arrays, the other one is containing the length of these arrays - check some properties ...

# file IO

- read files, write files
- write a function that gets a filename, return an array that has the content of the file (ints)
- write a function that gets two filenames, read the second half of the first file, write it into the second file
- compare file parameters...

# use the “extra.h” library

- #include “extra.h”
- use all the functions in there
- load a file, get all the counts for chars
- load two files, which letters are unique in file 1?

# pointers and structs

- use them in functions to allow for call by reference
- use them in basic contexts
- use them in conjunction with structs
- make functions that get structs as parameters
- make functions that return structs
- understand extra.h