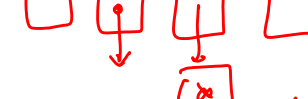


Final Exam 15-123 F09 Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, *** → 
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory *insufficient memory*
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [], ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

data types - int, double
storage classes
extern, auto, static, register

```
int foo() {
    extern int n;
}
```

```
malloc(sizeof(name))
```

```
int A[10];
int * ptr; *ptr = 4;
ptr[i] -> *(ptr+i)
```

```
int * ptr = malloc(100);
ptr[50] = 50;
*(ptr+50)
```

Perl Programming and Regex

1. Regex syntax, magnification(*,+,?), grouping, alternation
2. backreference
3. Starting with, not starting with

```
#define n 10
```

```
#define max(x,y) (x>y?x:y)
```

```
int x=10;
```

```
foo(x);
```

```
foo(int * ptr) {
    ...
}
```

/ a+b?c* /

/ (ab)+ /

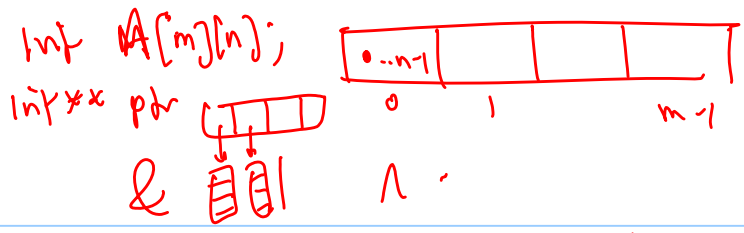
/ [0-9]? [a-b]* [a|b] /

/ a+(b?)c* /

↓
\$1

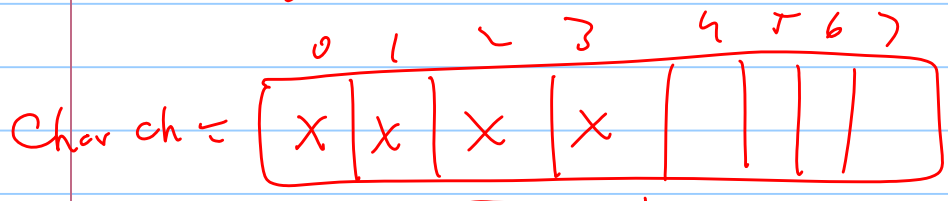
/ ^g /

/ ^ [^ a-z] /



```
int (fn*)(int, int);
```

fn = gcd;



$(ch \gg 4) \& 0x0F$

$x \wedge y = 0$

- Set bits to 1 $ch \mid 0xFO$
- Set bits to 0 $ch \& 0x0F$

```
CC = gcc
FLAGS = -ansi -Wall
```

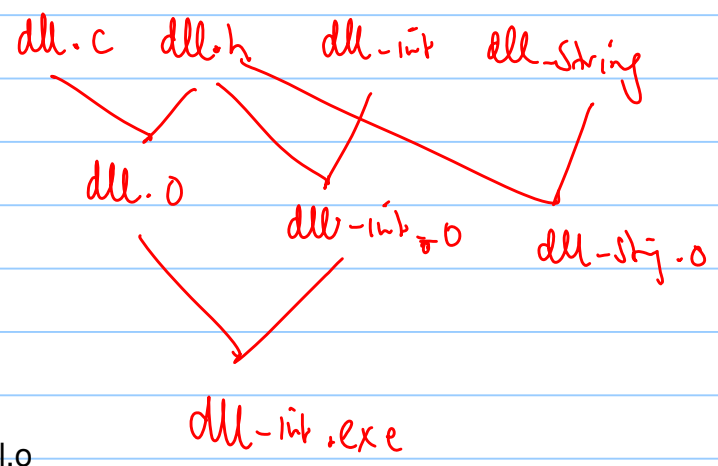
```
int : dll_int.o dll.o
$(CC) $(FLAGS) -o int dll_int.o dll.o
```

```
dll_int.o : dll_int.c dll.h
$(CC) $(FLAGS) -c dll_int.c
```

```
string : dll_string.o dll.o
$(CC) $(FLAGS) -o string dll_string.o dll.o
```

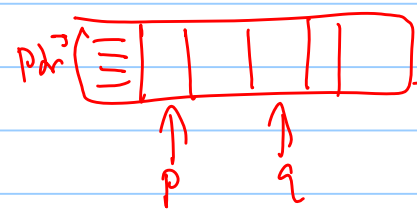
```
dll_string.o : dll_string.c dll.h
$(CC) $(FLAGS) -c dll_string.c
```

```
dll.o : dll.c dll.h
$(CC) $(FLAGS) -c dll.c
```



```
strcpy (*((Char**) p), ...)
```

```
void sort(ptr, blocks, block size, fn)
```



```
struct {
    char* name;
    int age;
} recd;
```

```
int comp(const void* p,
        const void* q) {
    (recd*)p -> name;
}
```

```
# defio int
typedef int money
```

```
foo(x)
```

```
} xptr = 10;
```

```
int *p; money x;
```

- 4. Groups [a-z] etc..
- 5. Binding (=) and negation (!)
- 6. Variables, if statements and loops
- 7. File open, read, write and close
- 8. System calls with perl

```
if (x == a+b?)
```

Shell Topics

- 1. Shell types. What shell are you on?
- 2. Writing a shell script, shell variables, environment, path, conditionals, loops
- 3. Command line arguments
- 4. Evaluating expressions in shell
- 5. Environment variables
- 6. Pipes



```
name="gun"
echo $name
```

```
echo $SHELL
$PATH
```

Miscellaneous

- 1. C and assembly
- 2. How programs get compiled
- 3. How programs get loaded and executed
- 4. Instruction Set Architecture
- 5. Hypothetical SAMS machine
- 6. Writing an assembly program for SAMS
- 7. Thinking about SAMS assembler and simulator

```
if [-d gun]
then
fi
```

```
$1, $2
```

```
'expr $1 + $2'
```

```
'ls -l | wc -l'
```

Saturday 1-3 TBA

→ Sunday - Email - Acedain

Final Exam 15-123 F09

Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, ***
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [] , ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

Perl Programming and Regex

1. Regex syntax, magnification(*,+?), grouping, alternation
2. backreference
3. Starting with, not starting with

4. Groups [a-z] etc..
5. Binding (=!) and negation (!~)
6. Variables, if statements and loops
7. File open, read, write and close
8. System calls with perl

Shell Topics

1. Shell types. What shell are you on?
2. Writing a shell script, shell variables, environment, path, conditionals, loops
3. Command line arguments
4. Evaluating expressions in shell
5. Environment variables
6. Pipes

Miscellaneous

1. C and assembly
2. How programs get compiled
3. How programs get loaded and executed
4. Instruction Set Architecture
5. Hypothetical SAMS machine
6. Writing an assembly program for SAMS
7. Thinking about SAMS assembler and simulator

Final Exam 15-123 F09

Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, ***
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [] , ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

Perl Programming and Regex

1. Regex syntax, magnification(*,+?), grouping, alternation
2. backreference
3. Starting with, not starting with

4. Groups [a-z] etc..
5. Binding (=!) and negation (!~)
6. Variables, if statements and loops
7. File open, read, write and close
8. System calls with perl

Shell Topics

1. Shell types. What shell are you on?
2. Writing a shell script, shell variables, environment, path, conditionals, loops
3. Command line arguments
4. Evaluating expressions in shell
5. Environment variables
6. Pipes

Miscellaneous

1. C and assembly
2. How programs get compiled
3. How programs get loaded and executed
4. Instruction Set Architecture
5. Hypothetical SAMS machine
6. Writing an assembly program for SAMS
7. Thinking about SAMS assembler and simulator

Final Exam 15-123 F09

Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, ***
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [] , ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

Perl Programming and Regex

1. Regex syntax, magnification(*,+,?), grouping, alternation
2. backreference
3. Starting with, not starting with

4. Groups [a-z] etc..
5. Binding (=!) and negation (!~)
6. Variables, if statements and loops
7. File open, read, write and close
8. System calls with perl

Shell Topics

1. Shell types. What shell are you on?
2. Writing a shell script, shell variables, environment, path, conditionals, loops
3. Command line arguments
4. Evaluating expressions in shell
5. Environment variables
6. Pipes

Miscellaneous

1. C and assembly
2. How programs get compiled
3. How programs get loaded and executed
4. Instruction Set Architecture
5. Hypothetical SAMS machine
6. Writing an assembly program for SAMS
7. Thinking about SAMS assembler and simulator

Final Exam 15-123 F09

Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, ***
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [] , ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

Perl Programming and Regex

1. Regex syntax, magnification(*,+?), grouping, alternation
2. backreference
3. Starting with, not starting with

4. Groups [a-z] etc..
5. Binding (=!) and negation (!~)
6. Variables, if statements and loops
7. File open, read, write and close
8. System calls with perl

Shell Topics

1. Shell types. What shell are you on?
2. Writing a shell script, shell variables, environment, path, conditionals, loops
3. Command line arguments
4. Evaluating expressions in shell
5. Environment variables
6. Pipes

Miscellaneous

1. C and assembly
2. How programs get compiled
3. How programs get loaded and executed
4. Instruction Set Architecture
5. Hypothetical SAMS machine
6. Writing an assembly program for SAMS
7. Thinking about SAMS assembler and simulator

Final Exam 15-123 F09

Review Sheet

This is a collection of topics that needs to be reviewed when preparing for the final exam. This is only a **subset** of what was covered in class. This sheet makes no implication that this is the only review material needed to prepare for the exam. You are still responsible for classnotes, labs, projects, homework etc etc..

Format: The exam is a 3 hour written exam given during the hub scheduled Final Exam time – Monday December 7th, 2009 from 1-4 PM. Room TBA. Exam will contain short questions from perl, shell, and C programming.

C programming topics

1. The meaning of a pointer, *, **, ***
2. Octal, decimal and hexadecimal
3. Arrays and pointers
4. Function and passing arguments. Why pass addresses?
5. Linked lists and hash tables - fundamentals
6. malloc, calloc and free
7. type of errors caused by a C program and how to detect it
 - a. unallocated memory
 - b. returning pointers to local variables
 - c. dereferencing uninitialized or unallocated memory
 - d. confusing pointer and reference types
 - e. array index out of bounds
 - f. double free of memory
8. Writing C macros and how they are processed?
9. macro vs typedef and macros vs functions
10. bit operations, getbit, setbit, masking
11. Missing libraries, what can go wrong?
12. Writing Make files and why write make files?
13. Finding memory leaks with valgrind. What causes memory leaks?
14. Debugging with gdb
15. Function pointers, typedef's and applications
16. void*'s and how to use them in developing generic libraries
17. using qsort, how to write a comparator
18. The difference between an array and a pointer? * vs [] , ** vs [][] etc..
19. Systems programming with C
 - a. File operations – open, creat, read, close, write, mkdir, rmdir etc
 - b. Process operations – fork, exec, wait/waitpid, exit,
20. Optimizing code – manually, with gcc flags

Perl Programming and Regex

1. Regex syntax, magnification(*,+,?), grouping, alternation
2. backreference
3. Starting with, not starting with

4. Groups [a-z] etc..
5. Binding (=!) and negation (!~)
6. Variables, if statements and loops
7. File open, read, write and close
8. System calls with perl

Shell Topics

1. Shell types. What shell are you on?
2. Writing a shell script, shell variables, environment, path, conditionals, loops
3. Command line arguments
4. Evaluating expressions in shell
5. Environment variables
6. Pipes

Miscellaneous

1. C and assembly
2. How programs get compiled
3. How programs get loaded and executed
4. Instruction Set Architecture
5. Hypothetical SAMS machine
6. Writing an assembly program for SAMS
7. Thinking about SAMS assembler and simulator