CSSE2310: 2016 Midsem exam answers

UQAttic

Get more out of your study time. Join UQAttic's revision chat.

Other exam papers

Please **contribute** to these documents.

> If you're looking for an effective way to familiarise yourself with the course material, you can't go past collaborating with fellow students. We have laboured to put these up, and so at the very least point out where you think we are wrong!
>
> You'll get more out of the course, you'll do better in the exam, and other students will benefit from your input as well.
>
> **To get editing permissions, simply go to the chatroom and provide us with your Google Account address.**

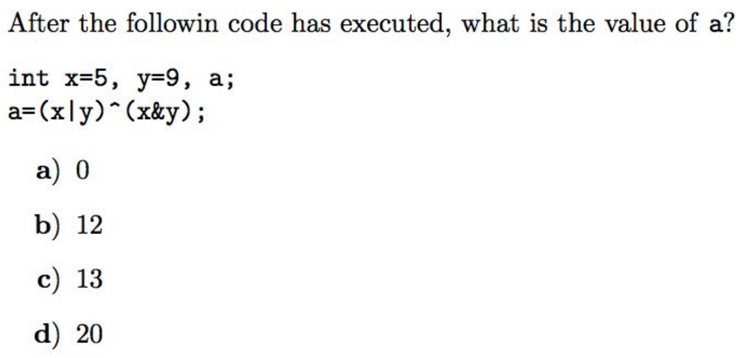
Style.

> Type answers in blue beneath each question.
>
> If you're unsure of your answer, highlight your answer text then hit Ctrl+Alt+M to create a comment beside the text. Once you're satisfied with the answer, click the "Resolve" button on the comment.
>
> [+x] Means x other contributors agreed with the solution.
>
> If you want some extra explanation from someone else on their answer, highlight the other person's answer and repeat the procedure above.

Communicate.

> Head over to uqattic.net and click "Chat Now!". You'll find a chatroom full of students just like you. Talk about a revision document (like this one) or swap prep tips. If you have your own IRC client, point it to irc.uqattic.net, port 6667, channel #attic.

Question 1



After the followin code has executed, what is the value of a?

```
int x=5, y=9, a;
a=(x|y)^(x&y);
```

a) 0

b) 12

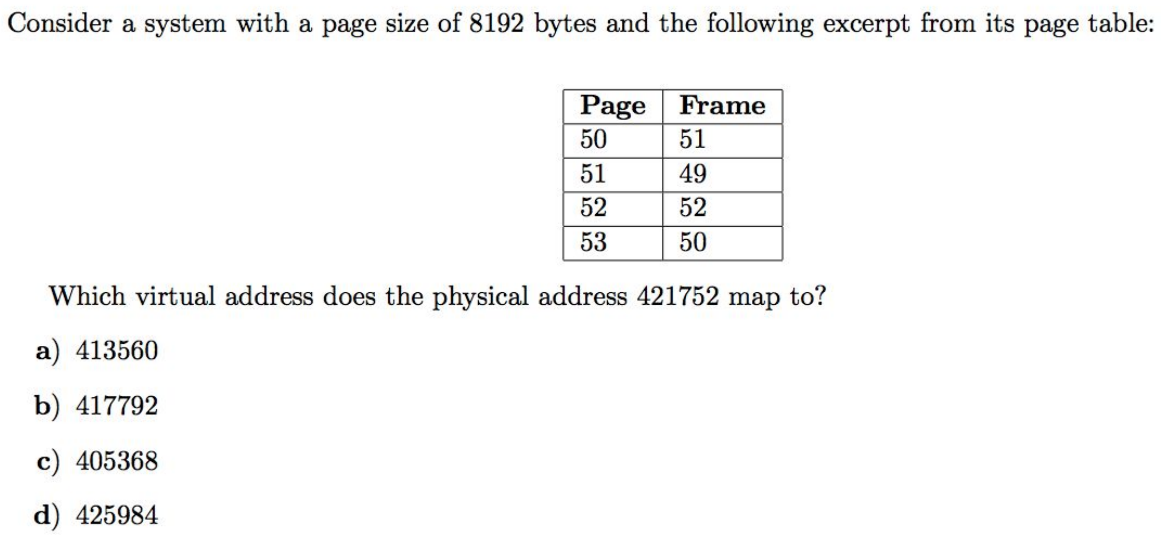
c) 13

d) 20

Bitwise operators: (x OR y) XOR (x AND y).
Convert x = 5; y = 9; to binary: 0101; 1001.
The bitwise OR is 1101
The bitwise AND is 0001
The bitwise XOR of these results is 1100
Convert to decimal: 12.

Answer: b) 12
[+4]

Question 2



Consider a system with a page size of 8192 bytes and the following excerpt from its page table:

Page	Frame
50	51
51	49
52	52
53	50

Which virtual address does the physical address 421752 map to?

a) 413560

b) 417792

c) 405368

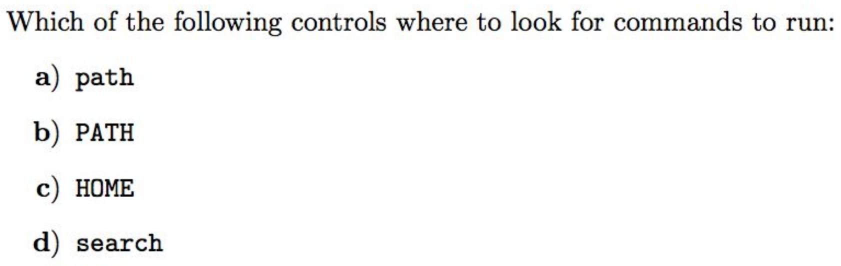
d) 425984

Convert physical address to Frame + Offset:
421752 / 8192 = 51 r3960
Frame 51 is mapped to page 50. Convert page to virtual address:
50 * 8192 = 409600

Add offset:
409600 + 3960 = 413560

Answer: a) 413560
[+4]

Question 3



Which of the following controls where to look for commands to run:

a) `path`

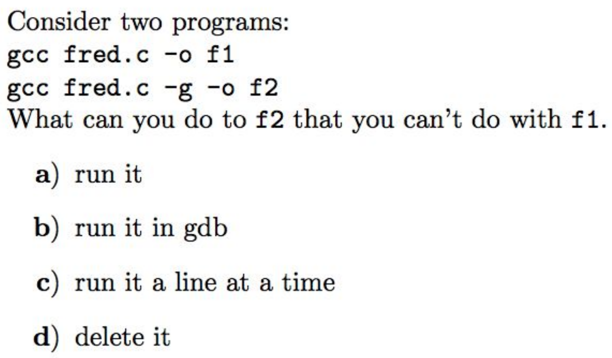
b) `PATH`

c) `HOME`

d) `search`

PATH – colon separated list of pathnames to search for commands
Answer: b)
[+4]

Question 4



Consider two programs:
`gcc fred.c -o f1`
`gcc fred.c -g -o f2`
What can you do to `f2` that you can't do with `f1`.

a) run it

b) run it in gdb

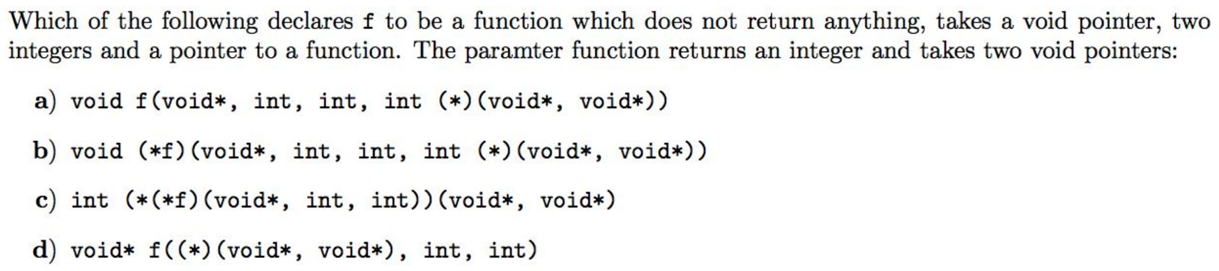
c) run it a line at a time

d) delete it

You can run an executable in gdb without having compiled with the -g flag BUT you can't step or see the code in gdb without the debug information in the executable.

Answer: c)
[+3]

Question 5



Which of the following declares `f` to be a function which does not return anything, takes a void pointer, two integers and a pointer to a function. The paramter function returns an integer and takes two void pointers:

a) `void f(void*, int, int, int (*)(void*, void*))`

b) `void (*f)(void*, int, int, int (*)(void*, void*))`

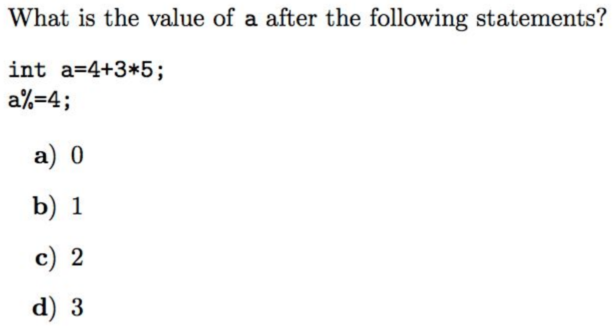
c) `int (*(*f)(void*, int, int))(void*, void*)`

d) `void* f((*)(void*, void*), int, int)`

f is a function which takes as arguments: a void pointer, two ints then a pointer to an unnamed function which takes two arguments (void pointers) and returns an int -- then (f) returns nothing.

Answer: a)
[+3]

Question 6



What is the value of `a` after the following statements?

```
int a=4+3*5;
a%=4;
```

a) 0

b) 1

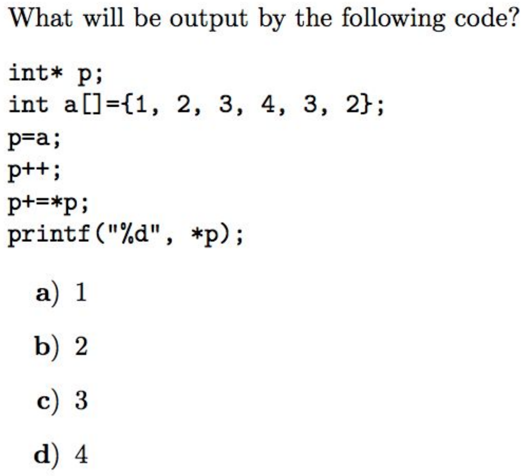
c) 2

d) 3

a = 4 + 15 == 19
19 % 4 == 3

Answer: d)
[+3]

Question 7



What will be output by the following code?

```
int* p;
int a[]={1, 2, 3, 4, 3, 2};
p=a;
p++;
p+=*p;
printf("%d", *p);
```

a) 1

b) 2

c) 3

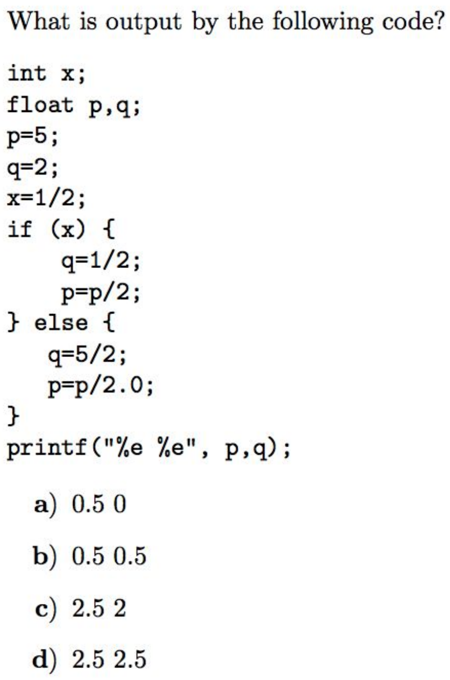
d) 4

Let (pn) denote the value of p on the n'th line.
'->' means points at.

a = [1, 2, 3, 4, 3, 2]
(p3) -> a[0]
(p4) = (p3) + 1 == &a[0] + 1 == &a[1]
(p4) -> a[1]
(p5) = (p4) + *(p4) == &a[1] + a[1] == &a[1] + 2 == &a[3]
(p5) -> a[3]
*(p5) == 4

Answer: d) 4
[+2]

Question 8



What is output by the following code?

```
int x;
float p,q;
p=5;
q=2;
x=1/2;
if (x) {
    q=1/2;
    p=p/2;
} else {
   q=5/2;
   p=p/2.0;
}
printf("%e %e", p,q);
```

a) 0.5 0

b) 0.5 0.5

c) 2.5 2

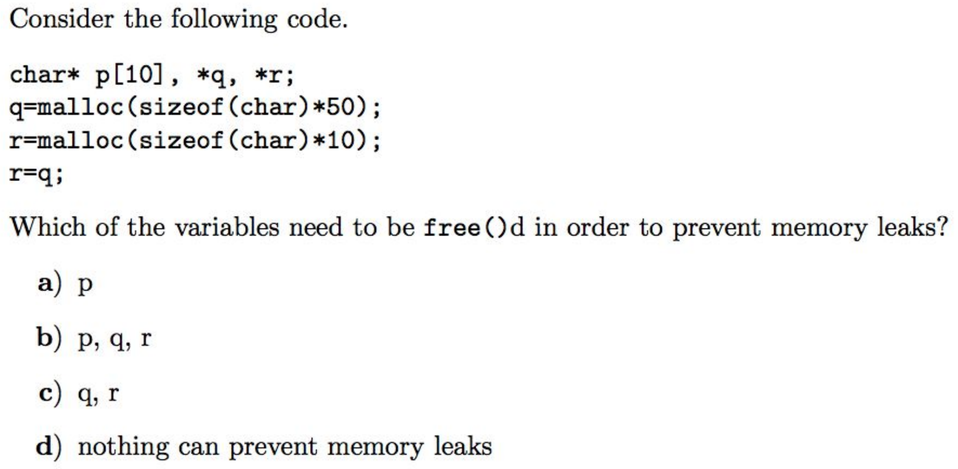
d) 2.5 2.5

After the 4th line,
p = 5.0, q = 2.0
The expression ½ on line 5 evaluates to 0. This is because both 1 and 2 are of type integers, and hence, truncation is performed during division (the remainder is thrown away).

Because x = 0, we go down the else path, where q is now assigned (5 / 2). Just as we reasoned above, this evaluates to 2 due to integer truncation. p is then equal to itself divided by 2.0, and since both are floats, 5.0/2.0 = 2.5

Hence: p = 2.5, q = 2.0
Answer: c) 2.5 2
[+3]

Question 9



Consider the following code.

```
char* p[10], *q, *r;
q=malloc(sizeof(char)*50);
r=malloc(sizeof(char)*10);
r=q;
```

Which of the variables need to be `free()`d in order to prevent memory leaks?

a) p

b) p, q, r

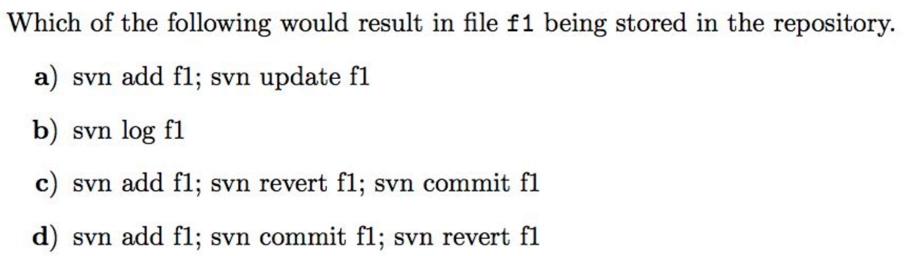
c) q, r

d) nothing can prevent memory leaks

On the 4th line of the code, when we assign the address of ‘q’ to ‘r’,
We lose our reference to the memory which was malloced to the original r, and because we have no way of referencing this memory, we cannot free it. It’s as lost as me during the function pointers lecture.

Answer: d)
[+2]

Question 10



Which of the following would result in file `f1` being stored in the repository.

a) svn add f1; svn update f1

b) svn log f1

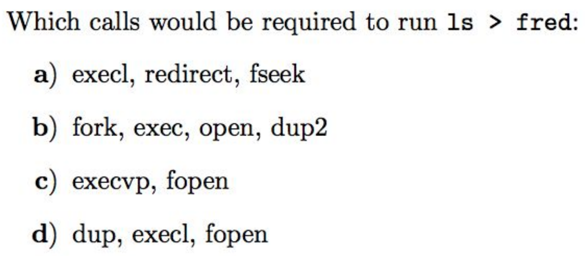
c) svn add f1; svn revert f1; svn commit f1

d) svn add f1; svn commit f1; svn revert f1

Can’t be A or B because there is no commit in either of them. If you reverted before committing, the file wouldn’t be in version control. Svn revert restores the working copy of the file to match the “clean” repo copy. Since you just committed f1, reverting it does nothing.

Answer: d)
[+3]

Question 11



Which calls would be required to run `ls > fred`:

a) execl, redirect, fseek

b) fork, exec, open, dup2

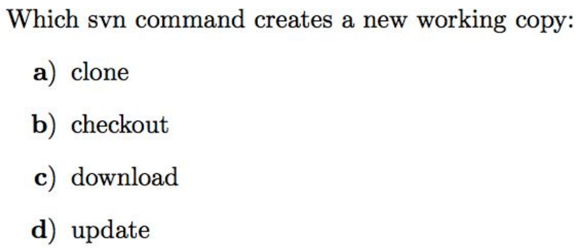
c) execvp, fopen

d) dup, execl, fopen

Option A isn’t a valid option since “redirect” isn’t the name of a syscall.
Option B seems to be the only valid option since it is the only one that forks the parent shell before executing. All the other options would terminate the user’s session because they didn’t fork.

Answer: b)
Answer b is correct. Tutors ran through this question in class and stated that fork or a similar process is essential
[+3]

Question 12



Which svn command creates a new working copy:

a) clone

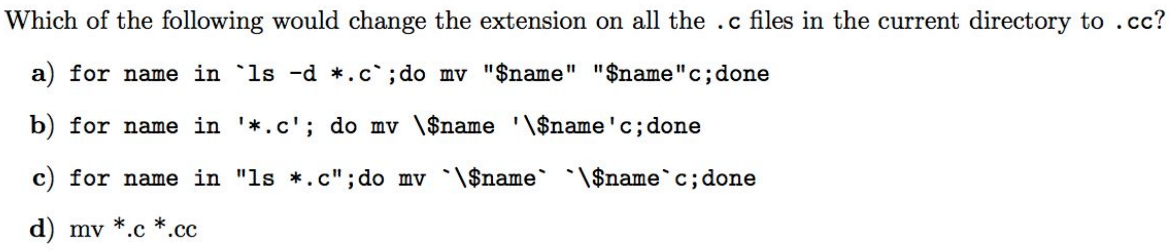
b) checkout

c) download

d) update

Answer: b) checkout
[+2]

Question 13



Which of the following would change the extension on all the `.c` files in the current directory to `.cc`?

a) `` for name in `ls -d *.c`;do mv "$name" "$name"c;done ``

b) `for name in '*.c'; do mv \$name '\$name'c;done`

c) `` for name in "ls *.c";do mv `\$name` `\$name`c;done ``

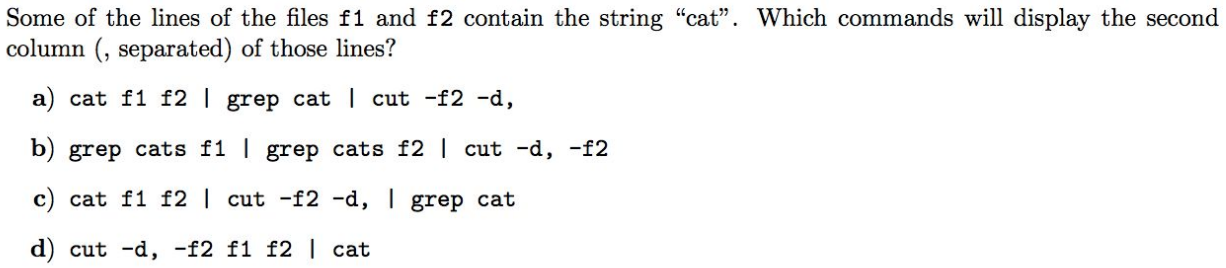
d) mv *.c *.cc

~~Ls -d *.c lists only directories with “.c” at the end, eliminating A. Ls isn’t even used in B, so no files were listed. The mv target can’t be ambiguous so it can’t be D. The remaining answer is C.~~
‘ls -d *.c’ will actually just list all the “.c” ending files. The ‘-d’ flag essentially prints out information for each of the directories that is passed as its arguments, just as it would if they were files. As such, the answer is A, as the backticks are a good indication this is also the

correct answer as this is the only one which would actually return the required output.

~~Answer: c)~~
Answer: a)
[+3]

Question 14



Some of the lines of the files `f1` and `f2` contain the string "cat". Which commands will display the second column (, separated) of those lines?

a) `cat f1 f2 | grep cat | cut -f2 -d,`

b) `grep cats f1 | grep cats f2 | cut -d, -f2`

c) `cat f1 f2 | cut -f2 -d, | grep cat`

d) `cut -d, -f2 f1 f2 | cat`

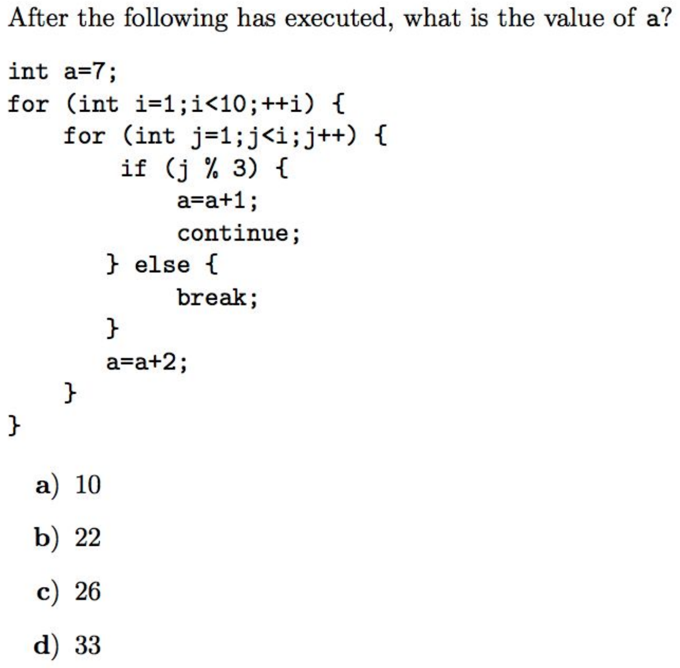
Option B can be eliminated since grep "cats" does not include the string we want "cat". Thus we need grep "cat".

Option D will cut every 2nd column separated by ',' in f1 and f2 without searching for cat first.
A & C 'cat f1 f2' will open files f1 and f2;

Option C will then cut every 2nd column separated string then will search for cat (grep cat) which is bad because unless that line in has cat in the 2nd column it will overlook any other 'cat' within the line.

Option A will first group all the lines with cat in it, then cut the 2nd column word and display those strings.
Answer: a) cat f1 f2 | grep cat | cut -f2 -d
[+2]

Question 15



After the following has executed, what is the value of **a**?

```
int a=7;
for (int i=1;i<10;++i) {
    for (int j=1;j<i;j++) {
        if (j % 3) {
            a=a+1;
            continue;
       } else {
            break;
       }
       a=a+2;
    }
}
```

a) 10

b) 22

c) 26

d) 33

Let us first inspect the inner for loop (to do with j)
We note that the ‘if’ statement tests if j is divisible by 3.

Remember that non zero statements evaluate to true.
That means any value of j not divisible by 3 will evaluate to true, and therefore enter the ‘if’ statement, where ‘a’ is incremented and the inner for loop starts again with j = j + 1.

If j is divisible by 3 (e.g. j = 3), then j % 3 = 0, and thus we will continue down the ‘else’ path, where we encounter a break statement which “breaks out of” the inner for loop. This means as soon as j = 3, we will break and move to the next value of i, meaning ‘a’ can be incremented a maximum of twice per loop.

The outer for loop will execute the inside code 9 times (10 - 1)
For i = 1, the inner for loop never executes (a += 0)
For i = 2, the inner for loop executes once (a += 1)
For all other values up to i = 9, of which there are 7, (9 - 2)
the inner for loop executes twice (a += 2)

So we add (0 + 1 + 7(2)) = 15 to the initial value of a.
Therefore a = 22 when the program terminates.

Answer: b) 22
[+2]