SENG265: Software Development Methods

Fall Term 2014

Mid-Term Exam

3:30 to 4:20 pm, Monday, 3 November 2014

Student Name: [redacted]

Student Number: V00 [redacted]

98

Marks: 22½ + 22 + 9½ + 27 + 17 out of 120

Important:

- This exam paper has 6 pages and contains 24 questions. Please check your copy of the paper and notify the instructor immediately if it does not include all 6 pages..
- All answers are to be written on this paper.
- This is a 50 minute exam.
- No aids are permitted. (Mute your cellphones and put them away!)

SECTION A: Linux Commands (5 points each)

For each question in this section, circle ALL answers which apply. (There may be more than one answer which applies.) Not circling anything is the same as selecting "None of the above".

Question 1: You are logged in as user `jj375` on a Linux account. Your current working directory, which has the path `/home/jj375/work` contains a file named `foo.sh`. When the command `ls -l foo.sh` is executed, it shows the following information for the file.

```
-rwxrw-r--  1  jj375  project9  11852  Feb 28 2014  foo.sh
```

Which of the following statements apply to this file?

(a.) The file is an executable program or script.

(b.) The file is more than 11 kilobytes in size.

c. Any user with an account on this computer can execute `foo.sh`.

(d.) Any user with an account on this computer can make a copy of the file.

e. None of the above.

5

ls -l

file mode | # links | owner | group | # bytes | modified | pathname

Question 2: The file `foo.sh` in the previous question can be accessed with which of these paths? (Recall that your current working directory is the directory which contains `foo`.)

a. ~~../foo.sh~~
b. `../work/foo.sh`
c. ~~`/jj375/home/work/foo.sh`~~
d. `~/../jj375/work/foo.sh`
e. None of the above

Question 3: Which of the following Linux commands could be used to make a copy of the file named `foo.sh` into another directory which has the path `/home/jj375/bin`? (Assume that your current working directory is still the directory which contains `foo.sh`.)

a. ~~`cat < foo.sh > ~/bin`~~
b. `cp foo.sh /home/jj375/bin`
c. `cat foo.sh | ~/jj375/bin/foo.sh`
d. ~~`cat > ~/bin/foo.sh < foo.sh`~~
e. None of the above.

Question 4: Which of these Linux commands will display the contents of a large text file named `bar.txt` without stopping?

a. `type bar.txt`
b. ~~`less bar.txt`~~
c. `list bar.txt`
d. `cat bar.txt`
e. ~~None of the~~ above.

Question 5: Which of these Linux commands will remove the file `bar.txt` from your home directory? (Circle an answer which removes the file regardless of what else the command may do.)

a. `rm ~/bar.txt`
b. `mv ~/bar.txt /tmp`
c. `rm ~/*`
d. `clear /home/bar.txt`
e. None of the above

SECTION B: Pointers in C (5 points each)

For each of the questions in this section, write your answer in the answer box. Each of these questions assumes the following declarations and initializations.

```
int k = 2;  int *p;  int a[] = { 1, 4, 9, 16 };
char *s = "SENG265";  char bb[100];
char *cp1, *cp2;
```

Question 6: What number is printed by this statement?

```
p = &k; printf("%d\n", a[*p]);
```

9

Question 7: What number is printed by this statement?

```
p = a; printf("%d\n", *(p+1));
```

4

Question 8: What number is printed by this code?

```
k = 0;
for(p=a; p<a+4; p++) {
   k += *p;
}
printf("%d\n", k);
```

30

Question 9: What string is printed by this code?

```
cp1 = bb;
for(k=0; k<4; k++) {
   *cp1++ = s[k++];
   *cp1++ = '*';
}
*cp1 = '\0';
printf("%s\n", bb);
```

S * E *

Question 10: What string is printed by this code?

```
cp2 = bb;
for(k=0; k<2; k++) {
   for(cp1 = s; *cp1 != '\0'; cp1++) {
      *cp2++ = *cp1++;
   }
}
*cp2 = '\0';
printf("%s\n", bb);
```

SN25SN25

SECTION C: Functions in C (5 points each)

Consider the file named `myfuncs.c` which contains the C code shown below.

```
// File: myfuncs.c
#include <stdio.h>
#include <stdlib.h>
#include "myfuncs.h"
static LSTP Head;
void Init() { Head = NULL; }
void Insert(double v) {
  LSTP n = malloc(sizeof(struct A));
  n->next = Head; n->val = v;
  Head = n;
}
LSTP GetHead() { return Head; }
void PrintList(void) {
  LSTP n;
  for(n=Head; n!=NULL; n=n->next) printf("%f\n", n->val);
}
```

Question 11: What *could* the identifier `LSTP` as used in `myfuncs.c` file be referring to?

a. `LSTP` may refer to a variable defined in `myfuncs.h`.

b. `LSTP` may refer to a datatype defined with a `typedef` in `myfuncs.h`.

c. `LSTP` may be a structure tag (i.e. it's used in a declaration `struct LSTP`) in `myfuncs.h`.

d. `LSTP` may be a C macro whose definition appears in `myfuncs.h`.

e. None of the above.

Question 12: What applies for the keyword `static` used in the declaration of `Head`?

a. `Head` is accessible by code in any file that forms part of the program.

b. Storage for `Head` is allocated in the heap area.

c. Statements in other files can copy the value of `Head` but cannot assign to `Head`.

d. The keyword `static` cannot be used here (the declaration of `Head` is erroneous).

e. None of the above.

Question 13: The `myfuncs.c` file contains a `#include` directive for `myfuncs.h` because ...

a. It lets the C compiler check that the functions declared in `myfuncs.c` are called correctly.

b. The `myfuncs.h` file provides the definition of LSTP.

c. The `myfuncs.h` file implements a library of functions that are useful in the `myfuncs.c` file.

d. The `myfuncs.h` file contains testing code for the functions declared in `myfuncs.c`.

e. None of the above.

SECTION D: Python 3 Coding (5 points each)

Write the output produced by each of the code snippets inside the box. Note: `"abc".join(a)` creates a string which consists of the elements of the tuple/list/sequence `a` using the string `"abc"` as the separator between consecutive elements.

Question 14:

```
a = [1, 3, 5]
b = a
a[1] = b[1]+1
print(a[1],b[1])
```

4 4

Question 15:

```
a = [1, 3, 5]
a[1] = [2,4]
print(' '.join(a))
```

1 [2,4] 5

Question 16:

```
a = [1, 3, 5, 7]
a[1:-2] = []
print(' '.join(a))
```

1 5 7 3

Question 17:

```
a = [1, 3, 5]
b = a[0:3]
a[0] = 2
print(' '.join(b))
```

1 3 5

Question 18:

```
def f(a):
    b = a
    return a+1
b = 5
print(b, f(2))
```

5 3

Question 19:

```
def f(a):
    global b
    b = a
    return a+1
b = 5
print(b, f(2))
```

3 2

Question 20:

```
for x in ("a", 3, [2,4]):
    print(x+x)
```

aa / 6 / [2,4,2,4] 2

SECTION D: Regular Expressions in Python (5 points each)

A subset of Python's metasymbols for regular expressions are shown in Table 1. You may use any of these metasymbols in your answers and NOT ANY OTHERS.

Table 1: Some Metasymbols Used in Regular Expressions

`.`	match any character except newline
`|`	separates two alternative
`(...)`	group elements of a subpattern
`\s`	match any white space character
`\w`	match a word character (letters, digits and underscores)

`\d`	match a digit character
`*`	repeat item 0 or more times
`+`	repeat item 1 or more times
`?`	repeat item 0 or 1 times
`[...]`	match one of the characters listed between the brackets

NO RANGE ?!

Question 21: Write a regular expression which matches a UVic id number. (For example `V00123456`; every id number begins with the same 3 characters and all have the same length.)

[V|v] 00 \d \d \d \d \d \d

I know it's UGLY

Question 22: Write a regular expression which matches a hexadecimal constant in C. (Two examples are `0x1a78F` and `0X00ee1`; every such constant has the prefix `0x` or `0X` and is followed by one or more hex digits which are 0 to 9 and a to f or A to F.)

0[x] [A-f a-F 0-9]+ 4

Question 23: Circle or underline the characters in the string below which match this regular expression, written as a raw string: `r"\w+\s*=\s*\w+\s*([-+*/]\s*\w+)*"`. The pattern does not have to match the whole string.

" cntr1 = cntr2 * 37 - x ; // do it! "

(ch whitespace word)* = (whitespace not included in this part)

* means 0 or more

will match cntr1 = cntr2 first, but also the * 37

Question 24: Write a pattern which matches a list of non-negative integers written in Python notation. (Two examples are `[]` and `[ 1, 3, 27,9 ]`.) Note that the brackets must be written as `\[` and `\]` inside the regular expression to remove the special meaning of these characters, and white space characters can appear, as the second example illustrates.

\[(\s*\d+[,])* \]

matches [3,] 3/5