

CS 60 — Spring 2009  
Quiz # 3 — Friday, May 8  
WRITE ALL YOUR ANSWERS ON SPACE PROVIDED  
ANSWER ALL QUESTIONS - TOTAL POINTS IS 50

NAME: \_\_\_\_\_

### Part 1

Circle if the statement is True (T) or False (F). Each question is worth 1 point. Assume that the content of the file `abc.txt` is as follows:

```
northwest  NW  Charles Main      3.0  .98  3  34
western    WE  Sharon Gray       5.3  .97  5  23
southwest  SW  Lewis Dalsass      2.7  .8   2  18
southern   SO  Suan Chin         5.1  .95  4  15
southeast  SE  Patricia Hemenway  4.0  .7   4  17
eastern    EA  TB Savage         4.4  .84  5  20
northeast  NE  AM Main Jr.       5.1  .94  3  13
north      NO  Margot Weber      4.5  .89  5   9
central    CT  Ann Stephens      5.7  .94  5  13
```

1. { **T** or F } The command `grep -i nw abc.txt` prints the line(s)

```
northwest  NW  Charles Main      3.0  .98  3  34
southeast  SE  Patricia Hemenway  4.0  .7   4  17
```

case insensitive

2. { T or **F** } The command `grep nw abc.txt` prints the line(s)

```
northwest  NW  Charles Main      3.0  .98  3  34
southeast  SE  Patricia Hemenway  4.0  .7   4  17 X
```

case sensitive

3. { **T** or F } The command `grep "^n" abc.txt` prints the line(s)

```
northwest  NW  Charles Main      3.0  .98  3  34
northeast  NE  AM Main Jr.       5.1  .94  3  13
north      NO  Margot Weber      4.5  .89  5   9
```

4. { T or **F** } The command `grep "5\.." abc.txt` prints the line(s)

```
western    WE  Sharon Gray       5.3  .97  5  23
southern   SO  Suan Chin         5.1  .95  4  15
northeast  NE  AM Main Jr.       5.1  .94  3  13
north      NO  Margot Weber      4.5  .89  5   9 X
central    CT  Ann Stephens      5.7  .94  5  13
```

5.x

5. { **T** or F } The command `grep "[A-Z] [A-Z] [A-Z]" abc.txt` prints the line(s)

```
eastern    EA  TB Savage         4.4  .84  5  20
northeast  NE  AM Main Jr.       5.1  .94  3  13
```

XYZ

6. { **T** or F } The command `awk '{print $3}' abc.txt` prints the line(s)
- ```
Charles
Sharon
Lewis
Suan
Patricia
TB
AM
Margot
Ann
```
7. { **T** or F } The command `awk '/Suan/' abc.txt` prints the line(s)
- ```
southern SO Suan Chin          5.1 .95 4 15
```
8. { T or **F** } The command `awk '/Charles/{print $5}' abc.txt` prints the line(s)
- ```
3.0
3.0 X
```
9. { **T** or F } The command `awk '/\./{print $2}' abc.txt` prints the line(s) **.9**
- ```
NW
WE
SO
NE
CT
```
10. { **T** or F } The command `awk '{printf "Name: %-8s and Place: %9s\n", $4, $1}' abc.txt` prints the line(s)
- ```
Name: Main      and Place: northwest
Name: Gray      and Place:  western
Name: Dalsass   and Place: southwest
Name: Chin     and Place:  southern
Name: Hemenway and Place: southeast
Name: Savage    and Place:   eastern
Name: Main     and Place: northeast
Name: Weber    and Place:   north
Name: Stephens and Place:  central
```

## Part 2

Circle the answer that you think is correct. Each question is worth 1 points.

- Given `int a, b=2`; What is `a`  
 (a) 0 (b) 1 (c) 2 **(d)** unknown
- Given `int a, b[5]={40,41,42,43}`; What is `b[2]`  
 (a) 40 (b) 41 **(c)** 42 (d) unknown
- Given `int a, b[5]={40,41,42,43}`; What is `b[4]` **unassigned**  
 (a) 41 (b) 42 (c) 43 **(d)** unknown
- Given `int a, b[5]={40,41,42,43}`; What is `b[5]` **undefined**  
 (a) 42 (b) 43 (c) 44 **(d)** unknown

5. Given `int a=5, *b=&a;` What is `*b` b points to a  
 (a) 0 (b) 5 (c) unknown address (d) unknown int
6. Given `int a=5, *b=&a;` What is `b`  
 (a) 0 (b) 5 (c) unknown address (d) unknown int
7. Given `int a[2]={5,6} *b=a;` What is `*b` b points to a[0]  
 (a) 5 (b) 6 (c) unknown address (d) unknown int
8. Given `int a[2]={5,6} *b=a;` What is `*(b+1)` b+1 points to a[1]  
 (a) 5 (b) 6 (c) unknown address (d) unknown int
9. Given `int a[2]={5,6} *b=a;` What is `*(b+2)` b+2 points to unknown  
 (a) 5 (b) 6 (c) unknown address (d) unknown int
10. Given `int a[2]={5,6} *b=a;` What is `(b+1)` b+1 is an address  
 (a) 5 (b) 6 (c) unknown address (d) unknown int

### Part 3

Write your answer in the white space. Questions are worth 5 points each.

1. Rewrite this code segment using `do .. while`

```
i = 0;
while(i<100) {
  sum += x[i];
  prod *= x[i];
  i++;
}
```

```
i=0;
do {
  sum += x[i];
  prod *= x[i];
  i++;
}
while(i<100);
```

2. What are the values of `i`, `j`, `k` at the end of the program for the values of `x` given below.

```
switch(x) {
  case 1: i++; break;
  case 2: j++;
  case 3: k++; break;
  case 4: i++;
  default: i=21; j=22; k=23;
}
```

|   | x     | 0  | 1 | 2 | 3 | 4  | 5  |
|---|-------|----|---|---|---|----|----|
| i | ----- | 21 | 1 | 0 | 0 | 21 | 21 |
| j | ----- | 22 | 0 | 1 | 0 | 22 | 22 |
| k | ----- | 23 | 0 | 1 | 1 | 23 | 23 |

3. After these declarations:

```
int i=3, j=5, *p = &i, *q = &j, *r;
double x;
```

What are the values of the following expressions?

Expression

Value

`p == (&i)`

1 p contains the address of i

`*(&p)`

3 \*(&p) is the same as p

`r = (&x)`

illegal because x of type double

`((7 * (*p))) / (*q) + 7`

11 (7\*3)/5 + 7 = 4+7 = 11

`(*(&j)) *= (*p)`

15 r points to j and j=j\*i = 5\*3 = 15

4. After these declarations:

```
int a[3][5]={{11,12,13,14,15},{21,22,23,24,25},{31,32,33,34,35}} ;
double x;
```

What are the values of the following expressions?

| Expression            | Value                 |
|-----------------------|-----------------------|
| a[0][1]               | a[0][1] : 12          |
| *(a[1] + 2)           | a[1][2] : 23          |
| *(a+2)[3]             | a[2][3] : 34          |
| *((*(a+2))+4)         | a[2][4] : 35          |
| *(&a[0][0] + 5*3 + 4) | a[3][4] : nonexistent |

5. After these declarations:

```
int i, j; int *p, *q;
```

Which of the following assignment expressions are legal?

| Expression    | Legal? |
|---------------|--------|
| p = &i        |        |
| p = &&i       |        |
| q = &p        |        |
| *q = &j       |        |
| p = q         |        |
| i = &j        |        |
| i = (*&)j     |        |
| i = *&&j      |        |
| i = (int) p   |        |
| i = *p++ + *q |        |

6. Explain what the following program prints.

```
#include <stdio.h>
```

```
int z;
```

```
void f(int x) {
    x = 2;
    z += x;
}
```

```
int main(void) {
    z = 5;
    f(z);
    printf("z = %d\n", z);
    return 0;
}
```

z = 7 is printed  
function f adds 2 to the global variable z  
without returning anything