CS 241
Data Organization
Midterm Postmortem

Brooke Chenoweth

University of New Mexico

Fall 2014

Question 1a

Which of the following are *not* keywords in the C programming language?

A boolean
B case
C continue
D do
E if
F int
G then
H type
I typedef
J union

Question 1a

Which of the following are *not* keywords in the C programming language?

A boolean

B case

C continue

D do

E if

F int

G then

H type

I typedef

J union

Question 1b

Which of the following are true?

A `0`

B `'0'`

C `'\0'`

D `1`

E `-1`

F `sizeof(char)-1`

G `4 >> 3`

H `0.25`

I `1/4`

J `1.0/4.0`

Question 1b

Which of the following are true?

A `0`

B `'0'`

C `'\0'`

D `1`

E `-1`

F `sizeof(char)-1`

G `4 >> 3`

H `0.25`

I `1/4`

J `1.0/4.0`

Question 2a

How many times is a do while loop guaranteed to loop?

A 0

B 1

C Infinitely

D Variable

Question 2a

How many times is a do while loop guaranteed to loop?

A 0

B 1

C Infinitely

D Variable

Question 2b

What is the final value of x when the following code is run?

```
int x;
for(x=0; x<10; x++) { }
```

A 10

B 9

C 0

D 1

E undefined

Question 2b

What is the final value of x when the following code is run?

```
int x;
for(x=0; x<10; x++) { }
```

A 10

B 9

C 0

D 1

E undefined

Question 2c

What is the return type of the function with the following prototype?

```
int foo(char x, float v, double t);
```

A char

B double

C float

D foo

E int

Question 2c

What is the return type of the function with the following prototype?

```
int foo(char x, float v, double t);
```

A char

B double

C float

D foo

E int

Question 2d

Which of the following is a proper declaration of a pointer?

A `int x;`

B `int &x;`

C `ptr x;`

D `int *x;`

Question 2d

Which of the following is a proper declaration of a pointer?

A `int x;`

B `int &x;`

C `ptr x;`

D `int *x;`

Question 2e

Which of the following gives the memory address of integer variable a?

A `*a`

B `a`

C `&a`

D `address(a)`

Question 2e

Which of the following gives the memory address of integer variable a?

A `*a`

B `a`

C `&a`

D `address(a)`

Question 2f

Which of the following gives the value stored at the address pointed to by pointer a?

A `a`

B `val(a)`

C `*a`

D `&a`

Question 2f

Which of the following gives the value stored at the address pointed to by pointer a?

A `a`

B `val(a)`

C `*a`

D `&a`

Question 3

```
1  #include <stdio.h>
2  int x=7;
3  int foo(int n)
4  { int y=5;
5    x += 3;
6    y -= 2;
7    n += x-y;
8    printf("foo: x=%d, y=%d, n=%d\n", x, y, n);
9    return n;
10 }
11 void main(void)
12 { int x, n;
13   n = 4;
14   x = foo(n);
15   printf("main: n=%d, x=%d\n", n, x);
16   x = foo(n);
17   printf("main: n=%d, x=%d\n", n, x);
18 }
```

Question 3

```
1  #include <stdio.h>
2  int x=7;
3  int foo(int n)
4  { int y=5;
5    x += 3;
6    y -= 2;
7    n += x-y;
8    printf("foo: x=%d, y=%d, n=%d\n", x, y, n);
9    return n;
10 }
11 void main(void)
12 { int x, n;
13   n = 4;
14   x = foo(n);
15   printf("main: n=%d, x=%d\n", n, x);
16   x = foo(n);
17   printf("main: n=%d, x=%d\n", n, x);
18 }
```

```
foo: x=10, y=3, n=11
main: n=4, x=11
foo: x=13, y=3, n=14
main: n=4, x=14
```

Question 4

```
1  #include <stdio.h>
2  void main(void)
3  { unsigned char x = 37;
4    unsigned char y = 62;
5    unsigned char z = 235;
6    unsigned char a = x << 3;
7    unsigned char b = x >> 3;
8    unsigned char c = x & y;
9    unsigned char d = x & z;
10   unsigned char e = x | y;
11   unsigned char f = x ^ y;
12   printf("a=%d\n", a);
13   printf("b=%d\n", b);
14   printf("c=%d\n", c);
15   printf("d=%d\n", d);
16   printf("e=%d\n", e);
17   printf("f=%d\n", f);
18 }
```

Question 4

```
1  #include <stdio.h>
2  void main(void)
3  { unsigned char x = 37;
4    unsigned char y = 62;
5    unsigned char z = 235;
6    unsigned char a = x << 3;
7    unsigned char b = x >> 3;
8    unsigned char c = x & y;
9    unsigned char d = x & z;
10   unsigned char e = x | y;
11   unsigned char f = x ^ y;
12   printf("a=%d\n", a);
13   printf("b=%d\n", b);
14   printf("c=%d\n", c);
15   printf("d=%d\n", d);
16   printf("e=%d\n", e);
17   printf("f=%d\n", f);
18 }
```

a=40

b=4

c=36

d=33

e=63

f=27

Question 5

```
1  #include <stdio.h>
2
3  void main(void)
4  {
5     char data[] = "testingTest";
6     char *linePt = &data[7];
7     data[2] = 'x';
8     *linePt = 'P';
9     printf("[%s], [%s]\n", data, linePt);
10 }
```

Question 5

```
1  #include <stdio.h>
2
3  void main(void)
4  {
5     char data[] = "testingTest";
6     char *linePt = &data[7];
7     data[2] = 'x';
8     *linePt = 'P';
9     printf("[%s], [%s]\n", data, linePt);
10 }
```

[textingPest], [Pest]

Question 6

```
1  #include <stdio.h>
2  struct Point { int x; int y; };
3  struct Point foo(struct Point p1, struct Point *p2)
4  { p1.x /= p2->y;
5    p2->x *= p1.y;
6    p1.y++;
7    p2->y--;
8    return p1;
9  }
10 void main(void)
11 { struct Point a = {9, 4};
12   struct Point b = {2, 3};
13   struct Point c = foo(a, &b);
14
15   printf("a=(%d, %d)\n", a.x, a.y);
16   printf("b=(%d, %d)\n", b.x, b.y);
17   printf("c=(%d, %d)\n", c.x, c.y);
18 }
```

Question 6

```
1  #include <stdio.h>
2  struct Point { int x; int y; };
3  struct Point foo(struct Point p1, struct Point *p2)
4  { p1.x /= p2->y;
5    p2->x *= p1.y;
6    p1.y++;
7    p2->y--;
8    return p1;
9  }
10 void main(void)
11 { struct Point a = {9, 4};
12   struct Point b = {2, 3};
13   struct Point c = foo(a, &b);
14
15   printf("a=(%d, %d)\n", a.x, a.y);
16   printf("b=(%d, %d)\n", b.x, b.y);
17   printf("c=(%d, %d)\n", c.x, c.y);
18 }
```

a=(9, 4)
b=(8, 2)
c=(3, 5)

Question 7 – Bad Code

```
1  #include<stdio.h>
2
3  int foo(float x);
4
5  void main(void)
6  {
7    int n=5;
8    printf("%d\n", foo(n));
9  }
10
11 int foo(int n)
12 {
13   return 2*n;
14 }
```

Question 7 – Bad Code

```
1  #include<stdio.h>
2
3  int foo(float x);
4
5  void main(void)
6  {
7    int n=5;
8    printf("%d\n", foo(n));
9  }
10
11 int foo(int n)
12 {
13   return 2*n;
14 }
```

Will fail to compile because of conflicting types for foo. Solution is to change prototype on line 3 to match the definition.

Question 8 – Bad Code

```
1  #include <stdio.h>
2
3  void printBinary(unsigned int n)
4  {
5    if (n / 2)
6    {
7      printBinary(n);
8    }
9    printf("%d", n % 2);
10 }
11
12 void main(void)
13 {
14   printBinary(31);
15 }
```

Question 8 – Bad Code

```
1  #include <stdio.h>
2
3  void printBinary(unsigned int n)
4  {
5    if (n / 2)
6    {
7      printBinary(n);
8    }
9    printf("%d", n % 2);
10 }
11
12 void main(void)
13 {
14   printBinary(31);
15 }
```

Will seg fault because of infinite recursion. Need to change argument to recursive call on line 7 from n to n/2

Question 9 – Bad Code

```
1  #include <stdio.h>
2
3  #define ARRAYSIZE = 10
4
5  void main(void)
6  {
7    int i;
8    int n[ARRAYSIZE];
9    n[0] = 1;
10   n[1] = 1;
11   for (i=2; i<ARRAYSIZE; i++)
12   {
13     n[i] = n[i-2] + n[i-1];
14   }
15   for (i=0; i<ARRAYSIZE; i++)
16   {
17     printf("%d ", n[i]);
18   }
19   printf("\n");
20 }
```

Question 9 – Bad Code

```
1  #include <stdio.h>
2
3  #define ARRAYSIZE = 10
4
5  void main(void)
6  {
7    int i;
8    int n[ARRAYSIZE];
9    n[0] = 1;
10   n[1] = 1;
11   for (i=2; i<ARRAYSIZE; i++)
12   {
13     n[i] = n[i-2] + n[i-1];
14   }
15   for (i=0; i<ARRAYSIZE; i++)
16   {
17     printf("%d ", n[i]);
18   }
19   printf("\n");
20 }
```

Will not compile because of the equals sign in the macro definition on line 3.

Question 10 – Bad Code

```
1  #include <stdio.h>
2
3  void printFactors(int n);
4
5  int main()
6  {
7    printFactors(36);
8    return 0;
9  }
10
11 void printFactors(int n)
12 {
13   int i = 2;
14   while(i < n);
15   {
16     if(n % i == 0) printf("%d ", i);
17     ++i;
18   }
19   printf("\n");
20 }
```

Question 10 – Bad Code

```
1  #include <stdio.h>
2
3  void printFactors(int n);
4
5  int main()
6  {
7    printFactors(36);
8    return 0;
9  }
10
11 void printFactors(int n)
12 {
13   int i = 2;
14   while(i < n);
15   {
16     if(n % i == 0) printf("%d ", i);
17     ++i;
18   }
19   printf("\n");
20 }
```

Run into an infinite loop on line 14 because the semicolon is an empty loop body.