CS32 Midterm 1 Study Guide Solutions

12. Given the statement, where a is a structure variable

    `*a.p`

    Choose the correct statement below:

A. The statement is equivalent to a->p
B. The statement will always produce a compile time error.
C. Will work if p, a pointer, is a member of the structure a, and points to accessible memory.
D. The statement should be written (*a).p
E. The compiler will adjust the order of operations based on the context.

13. Given the following program write the code for the two functions described below.

```
#include <iostream>
using namespace std;
int main ()
{
   const int NUMITEMS = 20;
   int nums[NUMITEMS] = {3, 2, 3, 3, 9, 6, 7, 5};
   int numItems=8;


   // deleteNum should delete all of the occurrences
   // of its 3rd argument from the array and return the
   // number of meaningful numbers left in the array. For
   // example nums after the call below start
   // {2, 9, 6, 7, 5, ...} and numItems 5
   numItems = deleteNum(nums, numItems, 3);

   // insertNum should insert its 3rd argument that
   // number of times into the array and return the
   // number of meaningful numbers in the array. For
   // instance nums after the call below should start
   // {2, 2, 2, 9, 6, 7, 5, ...} and numItems 7
   // You can assume there is enough space in the array.
   numItems = insertNum(nums, numItems, 2);

   return 0;
}
```

```
int deleteNum(int *nums, int numItems, int a) {
     int numItemsNew = 0;
     int *iter = nums;
     for (int i = 0; i < numItems; i++) {
          if (*iter != a) {
               *nums = *iter;
               nums++;
               numItemsNew++;
          }
          iter++;
     }
     return numItemsNew;
}

int insertNum(int *nums, int numItems, int times) {
     for (int i = numItems + times - 1; i >= times; i--)
          *(nums + i) = *(nums + i - times);
     for (int i = 0; i < times; i++)
          *(nums + i) = times;
     return numItems + times;
}
```

14. What will be the output of the program?

```
#include <iostream>
using namespace std;
int main ()
{
   int nums[] = {2, 4, 6, 1, 2, 3};
   int *p = nums;
   while (p < nums + 6)
   {
      if (*p % 3 == 0)
         *p = *p + 1;
      else if(*p % 2 == 0) {
         *p += 2;
         p = p + 1;
      }
      p = p + 1;
   }
   for (int i=0; i<6; i++) cout << nums[i] << " "; cout << endl;
   return 0;
}
```

4 4 7 1 4 3

15. What will be the output of the program?

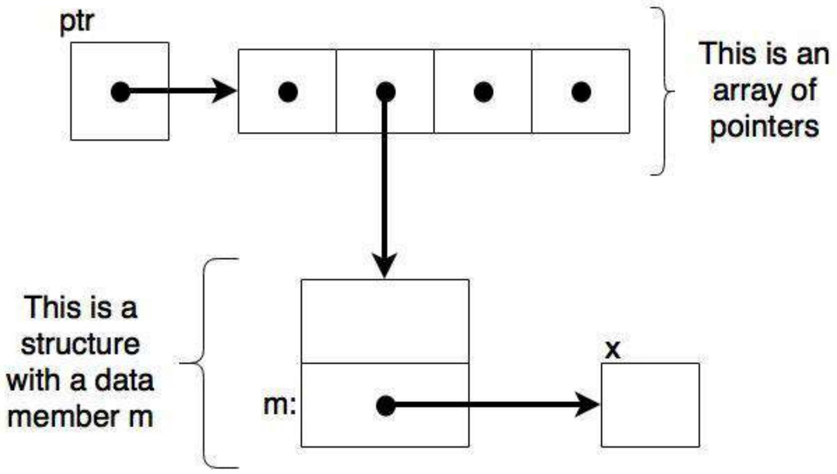
```
#include <iostream>
using namespace std;

int main ()
{
   int v1 = 10;
   int v2 = 25;
   int *p1 = &v1;
   int *p2 = &v2;
   *p1 += *p2;
   p2 = p1;
   *p2 = *p1 + *p2;
   cout << "*p1 = " << *p1 << " *p2 = " << *p2
        << " v1 = " << v1 << " v2 = " << v2 << endl;

   return 0;
}
```

***p1 = 70 *p2 = 70 v1 = 70 v2 = 25**

16. Write the code to set the variable x to 3 using ptr. You can assume the following diagram is already constructed in memory, hence you need only write one line of code.


ptr
This is an array of pointers
This is a structure with a data member m
m:
x

***(ptr[1] → m) = 3; OR *((*ptr[1]).m) = 3; OR *((*(ptr + 1)) → m) = 3;**

17. What will be the output of the program?

```
#include <iostream>
using namespace std;

void mystery (int* ptr, int& a, int& b);

int main ()
{
   int num1 = 0, num2 = 14, num3 = 17;
   int *p = &num1;
   mystery(p, num2, num3);
   cout << "num1: " << num1 << " num2: " << num2
        << " num3: " << num3 << endl;
   if (p == &num1) cout << " num1";
   else if (p == &num2) cout << " num2";
   else if (p == &num3) cout << " num3";

   return 0;
}

void mystery (int* ptr, int& a, int& b) {
   a++;
   b--;
   if (a > b)
      ptr = &a;
   else
      ptr = &b;
   *ptr += 10;

   return;
}
```

num1: 0 num2: 15 num3: 26
num1

18. Convert this infix expression to postfix: A + (B – C) / D   **A B C – D / +**

19. Evaluate the postfix expression: 5 4 3 + -   **-2**