CS 7B Midterm 2 - Spring 2018 - Midterm 2 5/7/18

Write responses to all questions on separate paper. Submit code by email, as appropriate.

1. Consider the problem of managing a 2D array on the heap in C++.
   (a) Write code to allocate a 1D array of 7 `int`s.
   (b) Write code to allocate a 2D 7 by 7 array of `int`s.
   (c) Write code to intialize your 2D array to all 1's.
   (d) Write code to allocate a 1D array of 49 `int`s.
   (e) Write a nested for loop to initialize the array of 7*7 `int`s as if it were a 2D array.
   (f) Write a function that will access the element in the ith row and jth column of your 1D array as if it were a 2D array.

2. Consider the following complete program:

```
1 #include <iostream>
  int main() {
3     int a = 17;
      double d1 = a, d2 = a;
5     std::cout << "\na␣=␣" << a << ",␣&a␣=␣" << &a;
      std::cout << "\nd1␣=␣" << d1 << ",␣&d1␣=␣" << &d1;
7     std::cout << "\nd2␣=␣" << d2 << ",␣&d2␣=␣" << &d2;
  }
```

   (a) How many bytes are allocated for `a`? For `d1`?
   (b) What is the output of this program on your computer?
   (c) This is an example of an implicit conversion. How could you make the conversion explicit?

3.
```
  #include <iostream>
2 int main() {
      int a = 17;
4     double d1 = *(double*)&a;
      std::cout << "\na␣=␣" << a << ",␣&a␣=␣" << &a;
6     std::cout << "\nd␣=␣" << d1 << ",␣&d␣=␣" << &d1;
  }
```

   (a) Describe how `&a` is an integer pointer.
   (b) What does `(double*)` do to `&a`?
   (c) The action on line 4 of this code is referred to as "type punning." Describe what is happening and how that explains the output you get on your machine. Have your written to unallocated memory? Have you read from unallocated memory?

4. Consider the following code:

```
1  #include <iostream>
   struct Thing {
3      int x, y;
       int* getPlaces() {
5          return &x;
       }
7  };
   int main() {
9      Thing t{8,15};
       int* place = (int*)&t;
11     std::cout << "\nt␣=␣" << place[0] << ",␣" << place[1];
       int y = *(int*)((char*)&t + 4);
13     std::cout << "\nt.y␣=␣" << y;
       int* location = t.getPlaces();
15     location[0] = 17;
       std::cout << "\nt.x␣=␣" << t.x;
17 }
```

   (a) Explain how line 9 affects your computer's memory.

   (b) Explain what place is and how it's initialized.

   (c) What is the result of the output on line 11? Why? Hint: These are raw memory operations: the kind of thing that C++ does really well.

   (d) What's going on on line 12? Explain.

   (e) Is there any copying of memory going on with line 14? If so, what is being copied? If not, why not? Explain.

5. Write a recursive method that uses only addition, subtraction, and comparison to multiply two numbers. The basic engine for this recursion is `multiply(n-1,m)+m;` where the base case returns m when $n - 1 = 1$. Be sure to handle the case where one or more factors is negative.

6. Write a recursive function to add the first n terms of the alternating harmonic series:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} \cdots$$