Name: Answers

CPSC 170A Test 2 Review

March 1st, 2017

1 Topics

- QT
- Pointers
- Searching
- Sorting
- Big-O

1. Perform a **good** handtrace for the following block of code.

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

#define SIZE 4

//Pre:  array is a 2-dimensional square array of size /size/.
//Post: prints the contents of array to the terminal.
void printArray(int * array[], int size) {
  for(int row = 0; row < size; row++) {
    for(int col = 0; col < size; col++) {
      cout << array[row][col] << " ";
    }
    cout << endl;
  }
}

//Pre:  array is a 2-dimensional square array of size /size/.
//Post: fills the array with some values.
void fillArray(int* array[], int size) {
  for(int row = 0; row < size; row++) {
    for(int col = 0; col < size; col++) {
      array[row][col] = (row + col) % 2;
    }
  }
}

//Pre:  array is a 1-dimensional array of empty pointers, of size /size/.
//Post: Makes array a 2-dimensional array of the specified size.
void createArray(int * array[], int size) {
  for(int i = 0; i < size; i++) {
    array[i] = new int[size];
  }
}

int main() {
  int* pointers[SIZE];
  createArray(pointers, SIZE);
  fillArray(pointers, SIZE);
  printArray(pointers, SIZE);
}
```

2. What is a memory leak? Are there any in the program from question 1.

A memory leak occurs when you leave dynamically allocated memory on the heap when you lose reference to it.
Technically, yes. Though pointers & memory it refers to gets removed @ the end of the program.

3. What does the Big-O class of a function tell you?

Big-O tells us how much longer our program runs, in the worst case, when the size of the input increases.

4. For the following snippets of code, write what would be printed to the terminal. If an error would occur, write **error**. If nothing is printed, write **nothing**.

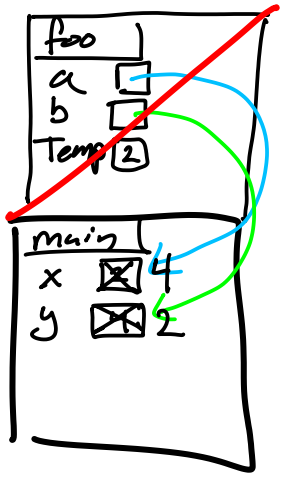
(a)
```
#include <iostream>
using namespace std;

void foo(int * a, int * b) {
  int temp = *a;
  *a = *b;
  *b = temp;
}

int main() {
  int x = 2;
  int y = 4;

  foo(&x, &y);

  cout << x << y << endl;
}
```



Output
42

(b)
```
#include <iostream>
using namespace std;
#define SIZE 4

int* createArray(int arraySize) {
  int array[arraySize];

  for(int i = 0; i < arraySize; i++) {
    array[i] = i + 1;
  }

  return array;
}

int main() {
  int * array = createArray(SIZE);

  for(int i = 0; i < SIZE; i++) {
    cout << array[i] << " ";
  }
  cout << endl;
}
```

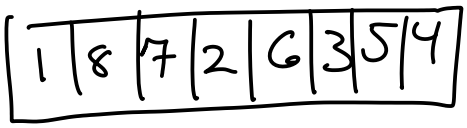
OH No! I'm trying to access memory that got deallocated!

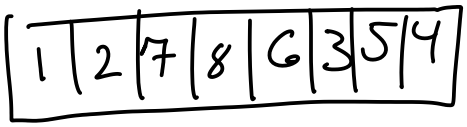
Seg Fault!!

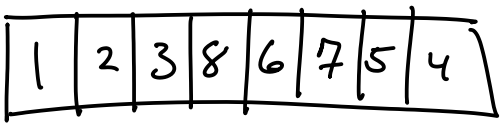
5. For the following array, show the state of the array after each iteration of the **Selection Sort** algorithm.

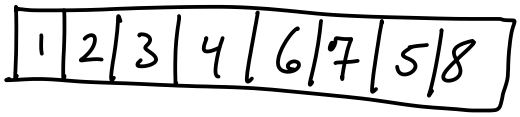
8	1	7	2	6	3	5	4

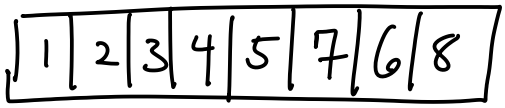
Min Selection

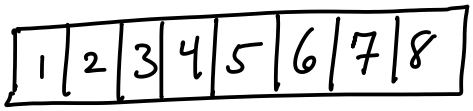




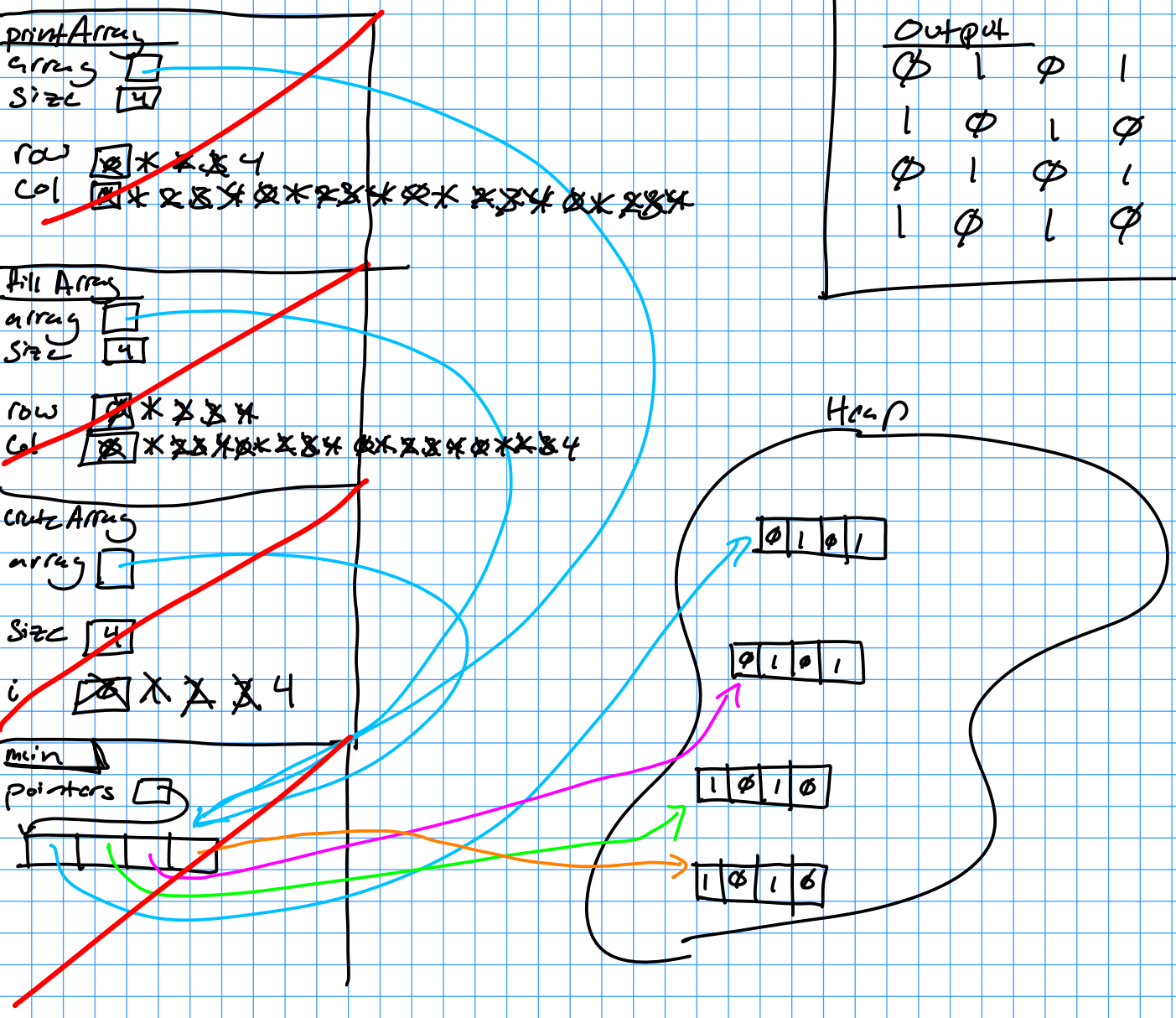








Hand Trace Q1



Output

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0