# 330 Midterm

Cole Pendergraft

1.)

a) Jeec can read/write

The group can read/write

Others can just read

b)

first malloc allocate

24 · 10 = 240 bytes

second malloc allocates

100 bytes

Third malloc allocates

50 bytes

390 bytes total

c) E

2)

a) B

b) F

c) E

3)

```
Void swap_arr(int** arr, int m) {
  for (int i=0; i<m; i++) {
    for(int j=0; j<m; j++) {
      if (i > j) {
        int tmp = arr[i][j];
        arr[i][j] = arr[m-i-1][j];
        arr[m-i-1][j] = tmp;
      }
    }
  }
```

4) init_2D(***array_2D, dim1, dim2){

struct struct_t** tmp = (struct struct_t**)
malloc(dim1 * sizeof(struct
struct_t*));

(all of that above is one line)

for(int i=0; i<dim1; i++){

3. tmp[i] = (struct struct_t**)
malloc(dim2 * sizeof(
STR_LEN));

}

4 *array = tmp;

}

Void free_2D(arr, Dim1, dim2){

for(int i=0; i<dim1; i++){

free(array[i]);

}

free(array);

}

5)

```
Void init_2d_grid(**arr, x, y, z, mass,
    num){
  Point_t** tmp = (point_t**) malloc
      (num * sizeof(Point_t*));
  for(int i=0; i<num; i++){
    tmp[i] = (point_t*) malloc
        (1 . sizeof(Point_t));
    tmp[i]->x = x[i];
    tmp[i]->y = y[i];
    tmp[i]->mass = mass[i];
    tmp[i]->force = 0.0;
    tmp[i]->val = (rand() % 10);
  }
*array = tmp;
}
```

(6)

```
Void Convert_1d(array_2d, rows, cols){
  int* 1d = (int*) malloc(rows*cols*
            sizeof(int));
  for(int i=0; i<cols; i++){
    for(int j=0; j<rows; j++){
      1d[j] = array_2d[j][i];
```

Out of time