

Do your answers on the answer sheets provided. When you write code, you do not need to have "include" or "using" statements.

Question 3

You are sorting the following vector:

186 121 73 79 192 138 16 68 198 131

In each part below, I am going to ask you what the vector looks like after a certain phase of a sorting algorithm:

- **Bubble:** Draw it after one pass of bubble sort.
- **Selection:** Draw it after two passes of selection sort.
- **Insertion:** Draw it after you sentinelize for insertion sort, and then do two passes.
- **Merge:** In the top level of merge sort, you make two recursive calls. Draw the vector after the first of these two recursive calls returns.
- **Bucket:** You are implementing the second bucket sort implementation as we did in class, which utilizes a vector twice the size of the original. Assume that the numbers are uniformly distributed between 0 and 200 (not including 200). Show me the (ten element) vector which you will sort with insertion sort at the end.

Choose your answers from the multiple choice below (they are in lexicographic order, so that it is easier to find your answer):

a.	16	68	73	79	121	131	138	186	192	198
b.	16	68	73	79	192	138	186	121	198	131
c.	16	68	73	121	192	138	186	79	198	131
d.	16	68	79	73	121	138	131	198	186	192
e.	16	68	79	73	131	138	121	198	186	192
f.	16	68	186	121	73	79	192	138	198	131
g.	16	73	79	121	192	138	186	68	198	131
h.	16	73	121	79	192	138	186	68	198	131
i.	16	73	121	186	79	192	138	68	198	131
j.	16	79	73	68	121	138	131	186	192	198
k.	16	79	73	68	121	138	131	198	186	192
l.	16	121	73	79	192	138	186	68	198	131
m.	16	186	121	73	79	192	138	68	198	131
n.	68	16	73	79	121	131	198	138	192	186
o.	68	16	73	79	121	138	192	186	198	131
p.	68	121	73	79	16	138	192	186	198	131
q.	73	79	121	186	192	16	68	131	138	198
r.	73	79	121	186	192	138	16	68	198	131
s.	73	121	186	79	192	138	16	68	198	131
t.	73	186	121	79	138	192	16	68	131	198
u.	121	73	79	138	16	68	131	186	192	198
v.	121	73	79	186	138	16	68	192	131	198
w.	121	73	79	192	138	16	68	198	131	186
x.	121	186	73	79	138	192	16	68	131	198
y.	121	186	73	79	192	138	16	68	198	131
z.	138	16	68	198	131	73	79	121	186	192

Question 4

Recall the heap-based priority queue data structure from lecture notes, which is reproduced to the right.

Implement the **Push()** method.

```
class PQueue {
public:
    PQueue();
    PQueue(vector <double> &v);
    void Push(double d);
    double Pop();
    int Size();
    int Empty();
    void Print();
protected:
    void Percolate_Down(int index);
    vector <double> h;
};
```