Worksheet 7 $\boxed{\text{Kaushal Banthia 19CS10039}}$

```
Define a max-bound = 1000.
typedef struct point
{
   int x, y ;
} point ;

void main ()
{
   point arr [max-bound ];
   int index = 0.
   //when we call Add, we do an assignment to index from its return value(*)
   // when we call Remove, we do an assignment ~~for~~ to index from ——"—— (*)
}
```

(*Look at bottom of next page */)

```
int ~~void~~ Add (point arr [ ], int ~~&~~ index ~~&~~ , int point_x, int point_y)
{
   if (index > max-bound) { printf(" Array full"); return -1; }
   else
   {
     arr [index ] . x = point_x;
     arr [index ]. y = point_y;
     return ~~index~~ (++ index);
   }

int ~~void~~ Remove (point arr [ ], int index , int point_x , int point_y)
{
   for(int i=0 ; i < index ; i++)
   {
     if (arr [i] . x == point_x && arr [i] . y = point_y)
     {
       for (int j = i ~~+1~~; j<(index-1); j++)
       {
         arr [j]. ~~x~~ x = arr [j+1] . x ;
       }  arr[j].y = arr [j+1] . y ;
```

Page No.
Date :

Worksheet 7 Kaushal Banthia 19CS10039

```
        return (--index);
        ~~break;~~
      } //end_if
    } //end_for
  }

void find farthest (point* arr[], int point_x, int point_y, int index)
{  initialise_heap by name heap[] => Array of structures of type point
   We can call heapify on the array, & where the condition in
   the heapify ~~is that abs (point arr[parent].x~~
   is that
   => abs ((heap[parent].x - point_x) - (heap[parent].y - point_y))
                 is ~~greater lesser than~~ greater than
   abs ((heap[i].x - point_x) - (heap[i].y - point_y))

   where parent = (i-1)/2 if we start heap from 0th index

    If this condition is true then its fine else, we swap the
     the parent of i with element at i.

   Then after this heapify, we just print heap[0].x and heap[0].y
   => printf("%d %d", heap[0].x, heap[0].y);

}
```

★ /* example: index = Add (arr, index, 5, 6)
   Example: index = Remove (arr, index, 5, 6)