# Programming Techniques 1 2012 exam

4 A. i.

```
void Search (int low, int high, String Str) {
    assert (is_ordered (low, high));

    int i = low-1;
    int j = high+1;

    while ( i+1 != j )
    {
      mid =  (i+j)/2;
      if  (le[arr[mid], Str)){
        i = mid; }
      else {
        j = mid;
      }
    index = i;
    found = (low <= index) && (Key.eq(d)(arr[index]));
} // done
```

ii)

```
void Insert (String Str, int pos, int low, int high) {
    For (int k = high+1; k > pos; k--){
        arr[k] = arr[k-1]
      arr[pos] = Str;
    }
}
```

iii)

```
void Insert Sort (int low, int high)
    for (int k = low+1; k <= high; k++) {
        Search (low, k-1, arr[k]);
        Insert (arr[k], index+1, low, k-1);
    }
```

2012 Programming Techniques Exam Part

| now how | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| | H | B | Ar | K | CA | NA | LI | SI | P | U |

Scan from left Arr [4]

Scan from right Arr [4]

Swap arr [0] and arr [4]

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| CA | B | Ar | K | H | NA | LI | SI | P | U |

Now sort arr [0] to arr [3]

and arr [5] to arr [9]

c.

```
void qSort(int left, int right) {
    String pivot;
    int i, j;
    pivot = arr[ (left+right)/2 ];
    partition (left, right, pivot
    i = L;
    j = R;
    if (left < j) { qSort (left, j) }
    if (i < right) { qSort (i, right) }
} // done
```

INFORMATION SYSTEMS

2

2

c. I N F I N I T E L Y

Scan left until N arr[4];

Scan from right until L arr[8]

Swap arr[4] and arr[8]

→

I N F I L I T E N Y

Scan from left → T arr[6].

Scan from right → E arr[7]

Swap arr[6] and arr[7]

I N F I L I E T N Y

swap E and N

→

I E F I L I N T N Y

Result after 1st partition.

d.

```
void qSort(int left, int right){
    String pivot;
    int i, j;
    pivot = arr[(left+right)/2];
    partition (left, right, pivot);   i=L;  j=R;
    if (left < j) qSort(left, j)
    if (i < right) qSort ~~(right-i~~ i, right);
```