CS101 – Computer Programming
Quiz for Friday – 26 September 2014

Q1. Consider the following function 'merge(int T1[], int T2[], int N)' written by a student to merge the elements of an array. The student calls this function 'merge' from the main program as merge(A,B,n), where 'A' and 'B' are the arrays of size 'n'. The function 'merge' returns the call to the main program after execution.

```
void merge(int T1[], int T2[], int N){
  int k =0, l = N/2, j = 0;
  do{
    if (T1[k] > T1[l]){
      T2[j] = T1[k];
      k++;
    }
    else{
      T2[j] = T1[l];
      l++;
    }
    j++;
  }while( k!=N/2 && l!=N);

  while(k < N/2){
    T2[j]=T1[k];
    j++;
    k++;
  }
  while(l < N){
    T2[j]=T1[l];
    j++;
    l++;
  }
}
```

Select the choice(s) which is/are appropriate:

A. The above function 'merge' will merge the unsorted arbitray values of array 'A' and 'B' in descending order and store the final result in B

B. The above function 'merge' will merge the unsorted arbitrary values of array 'A' and 'B' in ascending order and store the final result in B.

C. If the elements of the array 'A' from index '0' to 'n/2' are in descending order, and from index '(n/2)+1' to 'n-1' are in ascending order, then the final contents of the array B will be in descending order.

D. If the elements of the array 'A' from index '0' to 'n/2' are in ascending order, and from index '(n/2)+1' to 'n-1' are in descending order, then the final contents of the array B will be in descending order.

E. None of these

Refer Q2. On the next page

Q2. Consider the following part of the program written by a student for sorting array in descending order using a slight change in the selection sort algorithm as shown below. Assume that the array A needs to be sorted and the function 'selectionsort' is called from the main program in the form 'selectionsort(A,n)', where 'A' is an array of size 'n'.

```
int findIndexOfMax(int A[],int start, int end){
  int i, currMaxIndex = start;
  for(i = start; i< end; i++){
    if(A[i] >= A[currMaxIndex]) { currMaxIndex = i; }
  }
  return currMaxIndex;
}
int findIndexOfMin(int A[],int start, int end){
  int i, currMinIndex = start;
  for(i = start; i< end; i++){
    if(A[i] <= A[currMinIndex]) { currMinIndex = i; }
  }
  return currMinIndex;
}
void swap(int A[], int index1, int index2){
  if(index1!=index2){
    int temp = A[index1];
    A[index1] = A[index2];
    A[index2] = temp;
  }
}
void selectionsort(int A[100], int n) {
  int mid = (n-1)/2;
  for(int i = 0; i <= mid; i++) {
    maxIndex = findIndexOfMax(A, i, n);
    minIndex = findIndexOfMin(A, 0, n-i);
    swap(A, i, maxIndex);
    swap(A, n-i-1, minIndex);
  }
}
```

Select an appropriate choice with respect to the above code for any arbitrary unsorted array ‘A’ as input to the function ‘selectionsort’:

A. The above code for selection sort will sort the entire array ‘A’ in descending order.

B. The above code for selection sort will sort the entire array ‘A’ in ascending order

C. The above code for selection sort will sort part of the array in ascending order and the other part in descending order

D. None of these