Workshop 4 – Exercises

1) Bubble Sort

```
#include <stdio.h>
void main()
{
	int i,j,temp;
	int score[10];
	for (i=0;i<10;i++) //Taking 10 scores input
		{
			printf("Enter score for student %d: ",i+1);
			scanf("%d",&score[i]);
		}
	for (i=0;i<10;i++) // Bubble sort
		{
			for(j=i+1;j<10;j++)
			if (score[i]<score[j]) //Sorting in descending
order
			{
				temp=score[i];
				score[i]=score[j];
				score[j]=temp;
			}
		}
	printf("The ranked scores are as follows: \n");
	for (i=0;i<10;i++) //Printing
			printf("%d. %d\n",i+1,score[i]);

}
```

2) Swapping the values of two variables

```
#include <stdio.h>


void main()
{
	int a, b;
	printf("Enter first value in variable a: ");
	scanf("%d",&a);
	printf("Enter second value in variable b: ");
	scanf("%d",&b);
	a+=b; //Sum of two values is stored in a i.e. a+b
	b=a-b; //b is now sum-b, i.e. a+b-b = a
	a=a-b; //a is now sum-b (b is already updated as a), so a=a+b-a = b
	printf("The value in variable a is: %d\n",a);
	printf("The value in variable b is: %d\n",b);
}
```

3) Sorting again (Challenge)

```
#include <stdio.h>

void cycle_sort(int [], int);

void main()
{
      int i,j,temp;
      int score[10];
      for (i=0;i<10;i++) //Taking 10 scores input
            {
                  printf("Enter score for student %d: ",i+1);
                  scanf("%d",&score[i]);
            }
      cycle_sort(score,10); //Passing array and size of array to
sorting function
      printf("The ranked scores are as follows: \n");
      for (i=0;i<10;i++) //Printing
                  printf("%d. %d\n",i+1,score[i]);

}

/* Cycle Sort algorithm */

// Function sort the array using Cycle sort
void cycle_sort(int arr[], int n)
{
    // count number of memory writes
    int writes = 0;
    int cycle_start;
    int i;
    int temp;

    // traverse array elements and put it to on
    // the right place
    for (cycle_start = 0; cycle_start <= n - 2; cycle_start++)
     {
        // initialize item as starting point
        int item = arr[cycle_start];

        // Find position where we put the item. We basically
        // count all smaller elements on right side of item.
        int pos = cycle_start;
        for (i = cycle_start + 1; i < n; i++)
            if (arr[i] > item)
                pos++;

        // If item is already in correct position
        if (pos == cycle_start)
```

```
            continue;

        // ignore all duplicate  elements
        while (item == arr[pos])
            pos += 1;

        // put the item to its right position
        if (pos != cycle_start) {
            temp=item; //Swapping
            item=arr[pos];
            arr[pos]=temp;
            writes++;
        }

        // Rotate rest of the cycle
        while (pos != cycle_start) {
            pos = cycle_start;
            // Find position where we put the element
            for (i = cycle_start + 1; i < n; i++)
                if (arr[i] > item)
                    pos += 1;

            // ignore all duplicate  elements
            while (item == arr[pos])
                pos += 1;

            // put the item to its right position
            if (item != arr[pos]) {
                temp=item; //Swapping
                item=arr[pos];
                arr[pos]=temp;
                writes++;
            }
        }
    }
}
```