printf("\t Items compared: [%d, %d] ", arr[j], arr[j+1]); bubble-sort.c if(arr[j]>arr[j+1]) #include<stdio.h> temp = arr[j]; #include<stdbool.h> arr[j] =arr[j+1]; #define arraySize 10 arr[j+1] =int arr[arraySize] = {91, 15, 85, 13, 29, 62, 42, 36, temp; 16, 53}; swapped = void header(); true: void footer(): printf("=> void arrayContent(); swapped (%d, %d)\n",arr[j], arr[j+1]); void arrayDuringSorting(); else printf("=> not int main() swapped\n"); { header(); if(!swapped) break; printf("\n"); } printf("\n\t| Array Content [BEFORE } SORTING]: \n\n\t\t"); arrayContent(); void arrayContent() arrayDuringSorting(); // Sorting method printf("\n\t| Array Content [AFTER SORTING]: \n\n\t\t"); for(int i=0; i<arraySize; i++) arrayContent(); printf("%d ", arr[i]); footer(); printf("\n"); printf("\n"); } return 0; } void header() void arrayDuringSorting() printf("\n { int temp; bool swapped = false; printf("\n\t\tData Structures and printf("\n\t| Array Content [DURING Algorithm"); SORTING]: \n"); printf("\n\tLesson: Sorting"); for(int i=0; i<(arraySize-1); i++) printf("\t\tTitle: Bubble Sort"); printf("\n ----printf("\n\t * Iteration (%d): ----"); ",(i+1)); } printf("["); for(int a=0; a<arraySize; a++)</pre> printf("%d ",arr[a]); void footer() printf("]\n"); { printf("\n swapped = false; for(int j=0; j<((arraySize-1)-i);</pre> printf("\n\t\t ~ Royland V. Pepaño ~"); j++)

printf("\n\t\t A 2nd Year BSIT student");

{

```
while(pos > 0 && arr[pos-
        printf("\n
                                                           1]>insert)
======\n");
                                                                             {
}
                                                                                      arr[pos] = arr[pos-1];
                                                                                      pos--:
                                                                                      printf("\t\t* %d was
insertion-sort.c
                                                           moved to arr[%d].\n", arr[pos], i);
#include<stdio.h>
                                                                             if(pos!=i)
#define arraySize 10
                                                                                      printf("\t\t* %d was
int arr[arraySize] = {91, 15, 85, 13, 29, 62, 42, 36,
                                                           inserted at arr[%d].\n", insert, pos);
16, 53};
                                                                                      arr[pos] = insert;
                                                                             }
                                                                    }
void header();
                                                           }
void footer();
void arrayContent();
void arrayDuringSorting();
                                                           void arrayContent()
int main()
                                                                    for(int i=0; i<arraySize; i++)
{
                                                                             printf("%d ", arr[i]);
        header();
                                                                    printf("\n");
        printf("\n");
                                                           }
        printf("\n\t| Array Content [BEFORE
SORTING]: \n\n\t\t");
                                                           void header()
        arrayContent();
        arrayDuringSorting(); // Sorting method
                                                           {
        printf("\n\t| Array Content [AFTER
                                                                    printf("\n
SORTING]: \n\n\t\t");
         arrayContent();
                                                           ======");
                                                                    printf("\n\t\tData Structures and
                                                           Algorithm");
        footer();
                                                                    printf("\n\tLesson: Sorting");
        printf("\n");
                                                                    printf("\t\tTitle: Insertion Sort");
        return 0;
                                                                    printf("\n ------
}
                                                           }
void arrayDuringSorting()
                                                           void footer()
        int insert;
        int pos;
                                                                    printf("\n -----
        printf("\n\t| Array Content [DURING
SORTING]: \n");
                                                                    printf("\n\t\t ~ Royland V. Pepaño ~");
        for(int i=1; i<arraySize; i++)</pre>
                                                                    printf("\n\t\t A 2nd Year BSIT student");
         {
                                                                    printf("\n
                  printf("\n\t * Iteration (%d): ", i);
                  printf("[ ");
                                                              ======\n");
                  for(int a=0; a<arraySize; a++)
                   printf("%d ",arr[a]);
                  printf("]\n");
                  insert = arr[i];
                                                           merge-sort.c
                  pos = i;
                                                           #include<stdio.h>
```

```
temp[index] = arr[i];
                                                                                            i+=1;
#define arraySize 10
                                                                                   }
                                                                                   else
void header();
                                                                                   {
void footer();
                                                                                            temp[index] = arr[j];
void arrayContent();
                                                                                            j+=1;
void arraySort(int[], int, int);
                                                                                   }
void arrayMerge(int[], int, int, int);
                                                                                   index++;
                                                                         }
int main()
{
                                                                         if(i>mid)
         int arr[] = {91, 15, 85, 13, 29, 62, 42, 36,
16, 53};
                                                                                   while(j<=end)
         header();
                                                                                   {
                                                                                            temp[index] = arr[j];
         printf("\n");
                                                                                            index++;
         printf("\n\t| Array Content [BEFORE
                                                                                            j++;
SORTING]: \n\n\t\");
                                                                                   }
         arrayContent(arr);
                                                                         }
         arraySort(arr, 0, arraySize-1);
                                                                         else
         printf("\n\t| Array Content [AFTER
                                                                         {
SORTING]: \n\n\t\t");
                                                                                   while(i<=mid)
         arrayContent(arr);
                                                                                            temp[index] = arr[i];
         footer();
                                                                                            index++;
         printf("\n");
                                                                                            i++;
         return 0;
                                                                                   }
}
                                                                         }
                                                                         k = start;
                                                                         while(k<index)
void arraySort(int arr[], int start, int end)
                                                                         {
                                                                                   arr[k] = temp[k];
         if(start<end)
                                                                                   k++;
         {
                                                                         }
                   int mid = (start + end)/2;
                                                               }
                   arraySort(arr, start, mid);
                   arraySort(arr, mid+1, end);
                   arrayMerge(arr, start, mid,
                                                               void arrayContent(int arr[])
end);
         }
                                                                         for(int i=0; i<arraySize; i++)
}
                                                                                   printf("%d ", arr[i]);
void arrayMerge(int arr[], int start, int mid, int end)
                                                                         printf("\n");
                                                               }
         int i = start;
         int j = mid+1;
         int k, index = start;
                                                               void header()
         int temp[arraySize];
                                                                         printf("\n
         while(i<=mid && j<=end)
                                                                                =");
                                                                         printf("\n\t\tData Structures and
                   if(arr[i]<arr[j])
                                                               Algorithm");
```

```
printf("\n\tLesson: Sorting");
                                                                              switch(choice)
        printf("\t\tTitle: Merge Sort");
        printf("\n -----
                                                                                      case 1:
   ----");
                                                                                                insert();
}
                                                                                                break;
                                                                                       case 2:
                                                                                                removed();
void footer()
                                                                                                break;
{
                                                                                      case 3:
                    -----
        printf("\n
                                                                                                display();
----");
                                                                                                break;
        printf("\n\t\t ~ Royland V. Pepaño ~");
                                                                                       case 4:
        printf("\n\t\t A 2nd Year BSIT student");
                                                                                                exit = false;
        printf("\n
                                                                                                break;
                                                                                      default:
       ======\n");
                                                                                                printf("\n\t |
                                                           ERROR: Invalid keyword.\n");
                                                                              }
queue.c
                                                                              if(exit==true)
                                                                                      printf("\n
#include<stdio.h>
#include<stdlib.h>
                                                                     ==\n");
#include<stdbool.h>
                                                                              else
                                                                                       footer();
#define size 10
                                                                    printf("\n");
                                                                    return 0;
void header();
                                                           }
void footer();
void insert();
                                                           void insert()
void removed();
void display();
                                                                    if(rear==size)
                                                                             printf("\n\t | WARNING:
int choice, item;
                                                           Queue reached its maximum capacity.\n");
int rear = 0;
                                                                    else
int front = 0;
                                                                    {
int queue[size];
                                                                              printf("\t | Enter a number to
                                                           insert: ");
int main()
                                                                              scanf("%d", &item);
                                                                              printf("\t | Position: %d,
        header():
                                                           Inserted Value: %d\n", rear, item);
        printf("\n");
                                                                              queue[rear++] = item;
                                                                    }
        bool exit = true;
                                                           }
        while(exit)
         {
                                                           void removed()
                  printf("\n\t\t\~ Queue Menu ~");
                                                           {
                  printf("\n\n\t\t1. Insert");
                                                                    if(front==rear)
                  printf("\n\t\t2. Remove");
                                                                              printf("\n\t | WARNING:
                  printf("\n\t\t3. Display");
                                                           Queue is empty.\n");
                  printf("\n\t\t4. Exit");
                                                                    else
                  printf("\n\n\t | Enter your
choice: ");
                                                                              printf("\t | Position: %d,
                  scanf("%d", &choice);
                                                           Removed Value: %d\n", front, queue[front]);
```

```
front++;
                                                            void arrayContent();
                                                            void arraySort(int[], int, int);
         }
}
                                                            int main()
void display()
                                                            {
                                                                     int arr[] = \{91, 15, 85, 13, 29, 62, 42, 36,
         if(front==rear)
                                                            16, 53};
                  printf("\n\t | WARNING:
                                                                     header();
Queue is empty.\n");
         else
                                                                     printf("\n");
         {
                                                                     printf("\n\t| Array Content [BEFORE
                  printf("\t | Queue Size:
                                                            SORTING]: \n\n\t\t");
%d\n\t\t ", rear);
                                                                     arrayContent(arr);
                  for(int i=front; i<rear; i++)</pre>
                                                                     arraySort(arr, 0, arraySize-1);
                           printf("\n\t | Position:
                                                                     printf("\n\t| Array Content [AFTER
%d, Value: %d", i, queue[i]);
                                                            SORTING]: \n\n\t\t");
                  printf("\n");
                                                                     arrayContent(arr);
         }
}
                                                                     footer();
                                                                     printf("\n");
void header()
                                                                     return 0;
{
                                                            }
         printf("\n
                                                            void arraySort(int arr[], int start, int end)
         printf("\n\t\tData Structures and
                                                                     int index = start;
Algorithm");
                                                                     int i, temp;
         printf("\n\tLesson: Stack & Queue");
                                                                     int pivot = arr[end];
         printf("\t Title: Queue");
                                                                     if(start<end)
         printf("\n -----
                                                                               for(i=start; i<end; i++)
}
                                                                                        if(arr[i]<=pivot)
void footer()
{
                                                                                                 temp = arr[i];
         printf("\n -----
                                                                                                 arr[i] =
----");
                                                            arr[index];
         printf("\n\t\t ~ Royland V. Pepaño ~");
                                                                                                 arr[index] =
         printf("\n\t\t A 2nd Year BSIT student");
                                                            temp;
         printf("\n
                                                                                                 index++;
      ======\n");
                                                                               }
}
                                                                               temp = arr[index];
                                                                               arr[index] = arr[end];
                                                                               arr[end] = temp;
quick-sort.c
                                                                               arraySort(arr, start, index-1);
                                                                               arraySort(arr, index+1, end);
#include<stdio.h>
                                                            }
#define arraySize 10
                                                            void arrayContent(int arr[])
void header();
void footer();
                                                                     for(int i=0; i<arraySize; i++)
```

```
printf("\n\t| Array Content [AFTER
         {
                  printf("%d ", arr[i]);
                                                            SORTING]: \n\n\t\t");
                                                                      arrayContent();
         }
         printf("\n");
}
                                                                      footer();
                                                                      printf("\n");
void header()
                                                                      return 0;
                                                            }
{
         printf("\n
                                                            void arrayDuringSorting()
         printf("\n\t\tData Structures and
                                                                      int min;
Algorithm");
                                                                      printf("\n\t| Array Content [DURING
         printf("\n\tLesson: Sorting");
                                                            SORTING]: \n");
         printf("\t\tTitle: Quick Sort");
                                                                      for(int i=0; i<(arraySize-1); i++)
         printf("\n -----
                                                                               printf("\n\t * Iteration (%d): ",
}
                                                            i+1);
                                                                               printf("[ ");
void footer()
                                                                               for(int a=0; a<arraySize; a++)
                                                                                printf("%d ",arr[a]);
{
         printf("\n -----
                                                                               printf("]\n");
----");
         printf("\n\t\t ~ Royland V. Pepaño ~");
                                                                               min = i;
         printf("\n\t\t A 2nd Year BSIT student");
                                                                               for(int j=i+1; j<arraySize; j++)</pre>
         printf("\n
                                                                               {
                                                                                        if(arr[j]<arr[min])</pre>
   ======\n");
                                                                                                 min = j;
                                                                               }
selection-sort.c
                                                                               if(min!=i)
                                                                                        printf("\t\tItems
#include<stdio.h>
                                                            Swapped: [%d, %d]\n", arr[i], arr[min]);
                                                                                        int temp = arr[min];
#define arraySize 10
                                                                                        arr[min] = arr[i];
int arr[arraySize] = {91, 15, 85, 13, 29, 62, 42, 36,
                                                                                        arr[i] = temp;
16, 53};
                                                                               }
                                                                     }
void header();
                                                            }
void footer();
void arrayContent();
                                                            void arrayContent()
void arrayDuringSorting();
                                                                      for(int i=0; i<arraySize; i++)
int main()
{
                                                                               printf("%d ", arr[i]);
         header();
                                                                      printf("\n");
         printf("\n");
                                                            }
         printf("\n\t| Array Content [BEFORE
SORTING]: \n\n\t\t");
                                                            void header()
         arrayContent();
         arrayDuringSorting(); // Sorting method
```

```
printf("\n
                                                           void arrayDuringSorting()
=======");
        printf("\n\t\tData Structures and
                                                                    int inner, outer;
Algorithm");
                                                                    int insert;
        printf("\n\tLesson: Sorting");
                                                                    int interval = 1:
        printf("\t\tTitle: Selection Sort");
                                                                    int elements = arraySize;
                   -----
        printf("\n
                                                                    int i = 0;
                                                                    printf("\n\t| Array Content [DURING
                                                           SORTING]: \n");
                                                                    while(interval<=(elements/3))
void footer()
                                                                             interval = interval * 3 + 1;
                                                                    while(interval>0)
        printf("\n
                    -----
                                                                    {
                                                                             printf("\n\t * Iteration (%d): ",
        printf("\n\t\t ~ Royland V. Pepaño ~");
                                                           i+1);
        printf("\n\t\t A 2nd Year BSIT student");
                                                                              printf("[ ");
        printf("\n
                                                                             for(int a=0; a<arraySize; a++)
                                                                               printf("%d ",arr[a]);
                       _____
      ======\n");
                                                                              printf("]\n");
}
                                                                             for(outer=interval;
                                                           outer<elements; outer++)
shell-sort.c
                                                                             {
                                                                                       insert = arr[outer];
#include<stdio.h>
                                                                                      inner = outer;
                                                                                      while(inner>(interval-
#define arraySize 10
                                                           1) && arr[inner-interval] >= insert)
int arr[arraySize] = {91, 15, 85, 13, 29, 62, 42, 36,
16, 53};
                                                                                                arr[inner] =
                                                           arr[inner-interval];
void header();
                                                                                                inner -=
void footer();
                                                           interval;
void arrayContent();
                                                                                                printf("\t\t*
void arrayDuringSorting();
                                                           %d was moved.\n", arr[inner]);
                                                                                      arr[inner] = insert;
int main()
                                                                                      printf("\t\t* %d was
{
                                                           inserted at arr[%d].\n", insert, inner);
        header();
                                                                             interval = (interval-1)/3;
        printf("\n");
                                                                             i++;
        printf("\n\t| Array Content [BEFORE
                                                                    }
SORTING]: \n\n\t\");
                                                           }
        arrayContent();
        arrayDuringSorting(); // Sorting method
                                                           void arrayContent()
        printf("\n\t| Array Content [AFTER
SORTING]: \n\n\t\t");
                                                                    for(int i=0; i<arraySize; i++)
         arrayContent();
                                                                             printf("%d ", arr[i]);
        footer();
         printf("\n");
                                                                    printf("\n");
         return 0;
                                                           }
}
```

```
void header()
{
       printf("\n
       printf("\n\t\tData Structures and
Algorithm");
       printf("\n\tLesson: Sorting");
       printf("\t\tTitle: Shell Sort");
       printf("\n ------
}
void footer()
       printf("\n ------
       printf("\n\t\t ~ Royland V. Pepaño ~");
       printf("\n\t\t A 2nd Year BSIT student");
       printf("\n
======\n");
}
```