**Bahria University**

**Software Engineering Department**



**Course: CSC-221 DATA STRUCTURES & ALGORITHMS**

**Term: FALL 2019, Class: BSE 3(B)**

**Assignment No:**

|  |  |
| --- | --- |
| **0** | **1** |

**Submitted By:**

**(Name) Qaiser Abbas (Reg. No.) 57245**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Submission Date** | | | | | | | |  |  |  |  |  |
| **3** | **1** | **/** | **1** | **0** | **/** | **1** | **9** |  |  |  |  |

**(Date: DD/MM/YY)**

**Submitted To:**

**Engr. Saniya Shaikh**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Max Marks: \_\_\_\_\_\_\_\_\_\_\_ Marks Obtained: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q 1: - Write an algorithm that finds the smallest and largest number in a list (an array) of n numbers.**

**Solution:**

class Program

{

//Assignment Task 1 completed via QuickSort

public static void Sort(int [] a, int n)

{

Sort(a, 0, n - 1);

}

private static void Sort(int [] a, int low, int up)

{

if (low>=up) // zero or one element in sub list

return;

int p = Partition(a, low, up);

Sort(a, low, p-1); // sort left sublist

Sort(a, p + 1, up); // sort right sublist

}

private static int Partition(int [] a, int low, int up)

{

int temp, i, j, pivot;

pivot = a[low];

i = low + 1; // moves from left to right

j = up; // moves from right to left

while (i <= j)

{

while (a[i] < pivot && i < up)

i++;

while (a[j] > pivot)

j--;

if (i < j) // swap a[i] and a[j]

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

i++;

j--;

}

else // found proper place for pivot

break;

}

// proper place for pivot is j

a[low] = a[j];

a[j] = pivot;

return j;

}

static void Main(string[] args)

{

int i, n;

int[] a = new int[20];

Console.Write("Enter the number of elements in array: ");

n = Convert.ToInt32(Console.ReadLine());

for (i = 0; i < n; i++)

{

Console.Write("Enter Elements " + (i + 1) + ":");

a[i] = Convert.ToInt32(Console.ReadLine());

}

Sort(a, n);

Console.WriteLine("Smallest number is:" + a[0] );

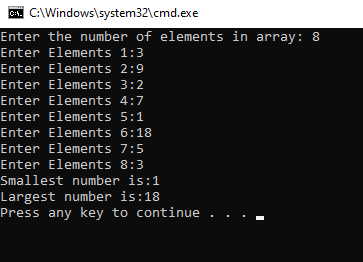
Console.WriteLine("Largest number is:" + a[n-1]);

}

}

}

**Output:**



**Q 2: - Write an algorithm that prints out all the subset of four elements of a set of n elements the elements of this set are sorted in a list that is input to the algorithm.**

**Solution:**

class Program

{

static void InputArray(int[] A)

{

for (int i = 0; i < A.Length; i++)

{

Console.WriteLine("Enter Integer value");

int.TryParse(Console.ReadLine(), out A[i]);

}

}

static void DisplaySubset(int[] set)

{

int count = 0;

for (int i = 0; i < set.Length; i++)

{

for (int j = i + 1; j < set.Length; j++)

{

for (int k = j + 1; k < set.Length; k++)

{

for (int l = k + 1; l < set.Length; l++)

Console.WriteLine("Subset # {0,-2}: [ {1} {2} {3} {4} ]", ++count, set[i], set[j], set[k], set[l]);

}

}

}

}

static void Main(string[] args)

{

Console.Write("Enter size of array: ");

int x;

int.TryParse(Console.ReadLine(), out x);

int[] data = new int[x];

InputArray(data);

Console.WriteLine("....All subsets of four elemnts of Entered array is:... ");

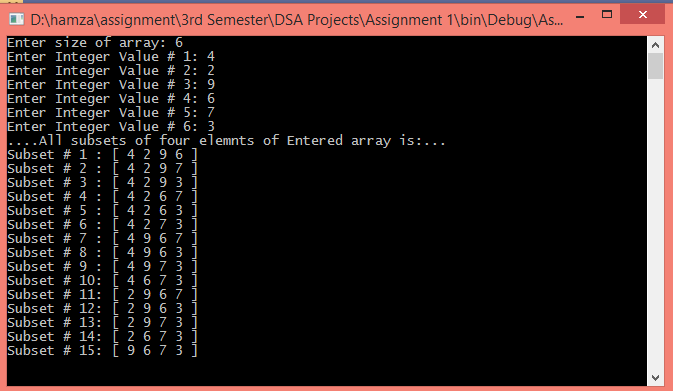
DisplaySubset(data);

}

}

}

**Output:**



**Q 3: - Design a program where user gives one string argument and generate result.**

Example:

* If input is “**abc**” then output should be “**klm**”.
* If input is “**xyz**” then output should be “**HIJ**”
* If input is “**gsm**” then output should be “**qCw**”
* If input is “**XYZ**” then output should be “**hij**”

**Solution:**

class Program

{

static void Main(string[] args)

{

string output;

string input;

Console.WriteLine("program where user gives one string argument and generate result");

Console.WriteLine("----------------------------------------------------------------");

Console.Write("Input the String:");

input = Convert.ToString(Console.ReadLine());

switch (input)

{

case "abc":

output = "klm";

break;

case "xyz":

output = "HIJ";

break;

case "gsm":

output = "qCw";

break;

case "XYZ":

output = "hij";

break;

default:

output = "Invalid string is provided";

break;

}

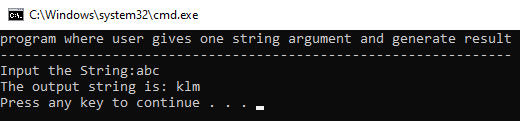
Console.Write("The output string is: {0}\n", output);

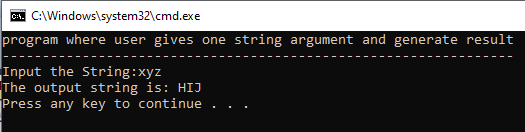
}

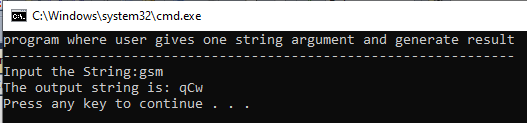
}

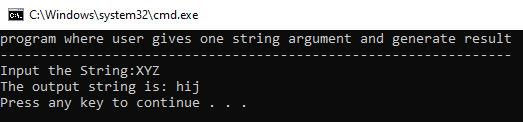
}

**Output:**









**Q 4: - Take 10 inputs from the user and assign them into two string arrays (make 2 unsorted string arrays of 5 lengths each), merge those arrays and obtain the result in the sorted manner.**

**Solution:**

class Program

{

public static void MergeSort(string[] Q, int lb, int ub)

{

int mid = (lb + ub) / 2;

if (lb < ub)

{

MergeSort(Q, lb, mid);

MergeSort(Q, mid + 1, ub);

Merge(Q, lb, mid, mid + 1, ub);

}

}

public static void Merge(string[] Arr, int lb01, int ub01, int lb02, int ub02)

{

int i, j, k;

string[] Copy = new string[Arr.Length];

for (i = 0; i < Arr.Length; i++)

{

Copy[i] = Arr[i];

}

for (i = lb01, j = lb02, k = lb01; i <= ub01 && j <= ub02 && k <= ub02; k++)

{

if (Copy[i].CompareTo(Copy[j]) < 0)

{

Arr[k] = Copy[i];

i++;

}

else

{

Arr[k] = Copy[j];

j++;

}

}

for (; i <= ub01; i++, k++)

{

Arr[k] = Copy[i];

}

for (; j <= ub02; k++, j++)

{

Arr[k] = Copy[j];

}

}

static void Main(string[] args)

{

string[] A = new string[5];

string[] B = new string[5];

Console.WriteLine("Enter 10 Random Strings:");

for (int i = 0; i < A.Length; i++)

{

Console.Write("[{0}] = ", i + 1);

A[i] = Console.ReadLine();

}

for (int j = 0; j < B.Length; j++)

{

Console.Write("[{0}] = ", j + 6);

B[j] = Console.ReadLine();

}

string[] C = new string[A.Length + B.Length];

for (int k = 0; k < A.Length; k++)

{

C[k] = A[k];

}

for (int l = 0; l < B.Length; l++)

{

C[l + 5] = B[l];

}

Console.WriteLine();

Console.WriteLine("New Sorted Array");

MergeSort(C, 0, C.Length - 1);

foreach (string item in C)

{

Console.Write(item + " ");

}

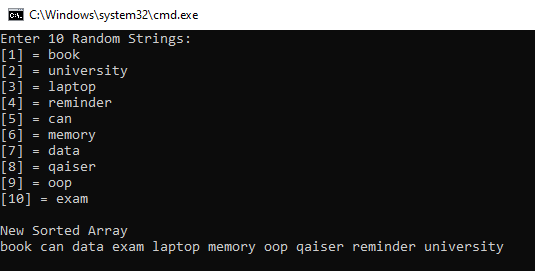
Console.ReadLine();

}

}

}

**Output:**



**Q 5: - Delete the string (taken from the user) from the array of strings that was inserted by the user. Use searching algorithm.**

**Solution:**

static void Main(string[] args)

{

Console.Write("Kindly Enter the length of String Array: ");

int x = int.Parse(Console.ReadLine());

string[] Q = new string[x];

Console.WriteLine("Enter values of String Array: ");

for (int i = 0; i < Q.Length; i++)

{

Console.Write("A[{0}] = ", i);

Q[i] = Console.ReadLine();

}

Console.WriteLine();

Console.WriteLine("Enter string you want Delete!");

string delete = Console.ReadLine();

Console.WriteLine("After deleting {0}, Now our new string array is: ", delete);

LinearSearch(Q, delete);

Console.ReadLine();

}

public static void LinearSearch(string[] A, string value)

{

for (int i = 0; i < A.Length; i++)

{

if (A[i] == value)

{

A = A.Except(new string[] { value }).ToArray();

}

}

foreach (string text in A)

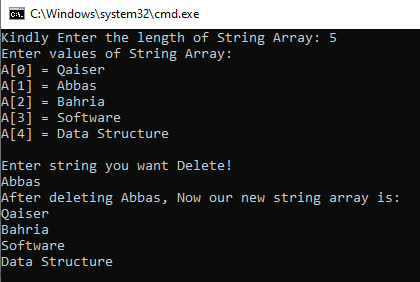
{

Console.WriteLine(text);

}

}

**Output:**



**Q 6: - Write a program which takes two input values from user and sums up all the even numbers between these numbers. Using recursion.**

**Solution:**

class Program

{

static int SumOfEvenNumbers(int num1, int num2, int sum = 0)

{

if (num1 > num2)

return 0;

if (num1 % 2 == 0)

{

Console.Write(num1 + " ");

return num1 + SumOfEvenNumbers(num1 + 1, num2);

}

else return SumOfEvenNumbers(num1 + 1, num2);

}

static void Main(string[] args)

{

Console.Write("Enter Number 1: ");

int num1;

int.TryParse(Console.ReadLine(), out num1);

Console.Write("Enter Number 2: ");

int num2;

int.TryParse(Console.ReadLine(), out num2);

Console.WriteLine("List of Even Numbers Between {0} and {1} is as: ", num1, num2);

Console.WriteLine("\nSum is: " + SumOfEvenNumbers(num1, num2));

}

}

}

**Output:**

