UCSC

University of Colombo School of Computing

IS1214 DataStructures and Algorithms

Practical 01

Try the following questions and create a pdf with all the screenshots of your outputs. Rename it as (IndexNo_Practical01)

Question 1

- Write a C program to insert an element at a specific position

Answer to the Question 1

```
#include <stdio.h>
int main() {
    int arr[100], newArr[101];
    int n, pos, x, i, j;

    // Get number of elements
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    // Input elements
    printf("Enter %d elements:\n", n);
    for (i= 0; i< n; i++) {
        scanf("%d", &arr[i]);
    }
```

Answer to the Question 1 Cont..

```
// Get element and position
   printf("Enter the element to insert: ");
   scanf("%d", &x);
    printf("Enter the position (0 to %d): ", n);
   scanf("%d", &pos);
    // Create new array with one extra element
   for (i= 0; i< pos; i++) {
       newArr[i] = arr[i];
   }
   // Insert new element
   newArr[pos] = x;
    // Copy remaining elements
      for (i= pos; i< n; i++) {
        newArr[i+ 1] = arr[i];
   }
```

Answer to the Question 1 Cont..

```
// Display new array
    printf("Array after insertion:\n");
    for (i= 0; i<= n; i++) {
        printf("%d ", newArr[i]);
    }

    printf("\n");


    return 0;
}
```

Question 2

- Write a C program to **search for an element in an array.**
  - The program should acceptthe size ofthe arrayandthe arrayelements from the user.
  - Then, it should take an element to search for and display whether the element is foundor not foundin the array.

Answer – Question 2

```
// C program to implement linear
// search in unsorted array
#include <stdio.h>
// Function to implement search operation
int findElement(int arr[], int n, int key)
{
   int i;
   for (i= 0; i< n; i++)
      if (arr[i] == key)
         return i;
   // If the key is not found
   return -1;
}
```

```
// Driver's Code
int main()
{
   int arr[] = { 12, 34, 10, 6, 40 };
   int n = sizeof(arr) / sizeof(arr[0]);
   // Using a last element as search element
   int key = 40;
   // Function call
   int position = findElement(arr, n, key);
   if (position == -1)
      printf("Element not found");
   else
      printf("Element Found at Position: %d",
           position + 1);
   return 0;
}
```

Question 3

- Write a C program to delete an element from an arrayat a given position.
  - The programshould acceptthe size of thearray and itselementsfromthe user.
  - Then, it should take the position of the element to be deleted.
  - After deletion, display the new array without the deleted element.

Answer - Question 3

```
#include <stdio.h>
 int main() {
    int n, pos, i, j = 0;
     // Input size of the array
    printf("Enterthenumberof elements in the array: ");
    scanf("%d", &n);
     intarr[n];
    intnewArr[n-1]; //Newarray will have one less element
```

Answer - Question 3 (Cont..)

```
// Input elements of the array
    printf("Enter %d elements:\n", n);
    for (i= 0; i< n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input position to delete
    printf("Enter the position to delete (1 to %d): ", n);

    scanf("%d",&pos);
```

Answer - Question 3 (Cont..)

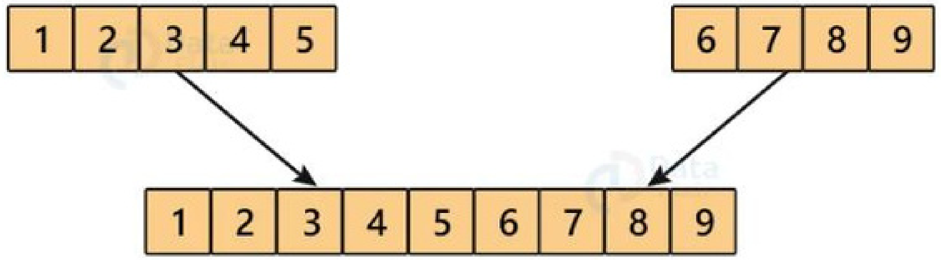
```
// Validate position
  if (pos < 1 || pos > n) {
    printf("Invalid position!\n");
    return 1;
  }
   // Copy all elements except the one at position 'pos'to new array
  for (i= 0; i< n; i++) {
    if (i!= pos -1) {
      newArr[j] = arr[i];
      j++;
    }
  }
```

Answer - Question 3 (Cont..)

```
// Display new array
  printf("Arrayafterdeletingelement at position %d:\n", pos);
  for (i= 0; i< n -1; i++) {
     printf("%d ", newArr[i]);
  }

  printf("\n");

  return 0;

}
```

Question 4

- Write an algorithm, pseudo code, and a C programto merge two sorted arrays into one array.
- Requirements:
- Accept the size and elements of two arrays from the user.
- Merge both arrays into a single array.
- Display the final merged array.
- E.g.



Answer question 4

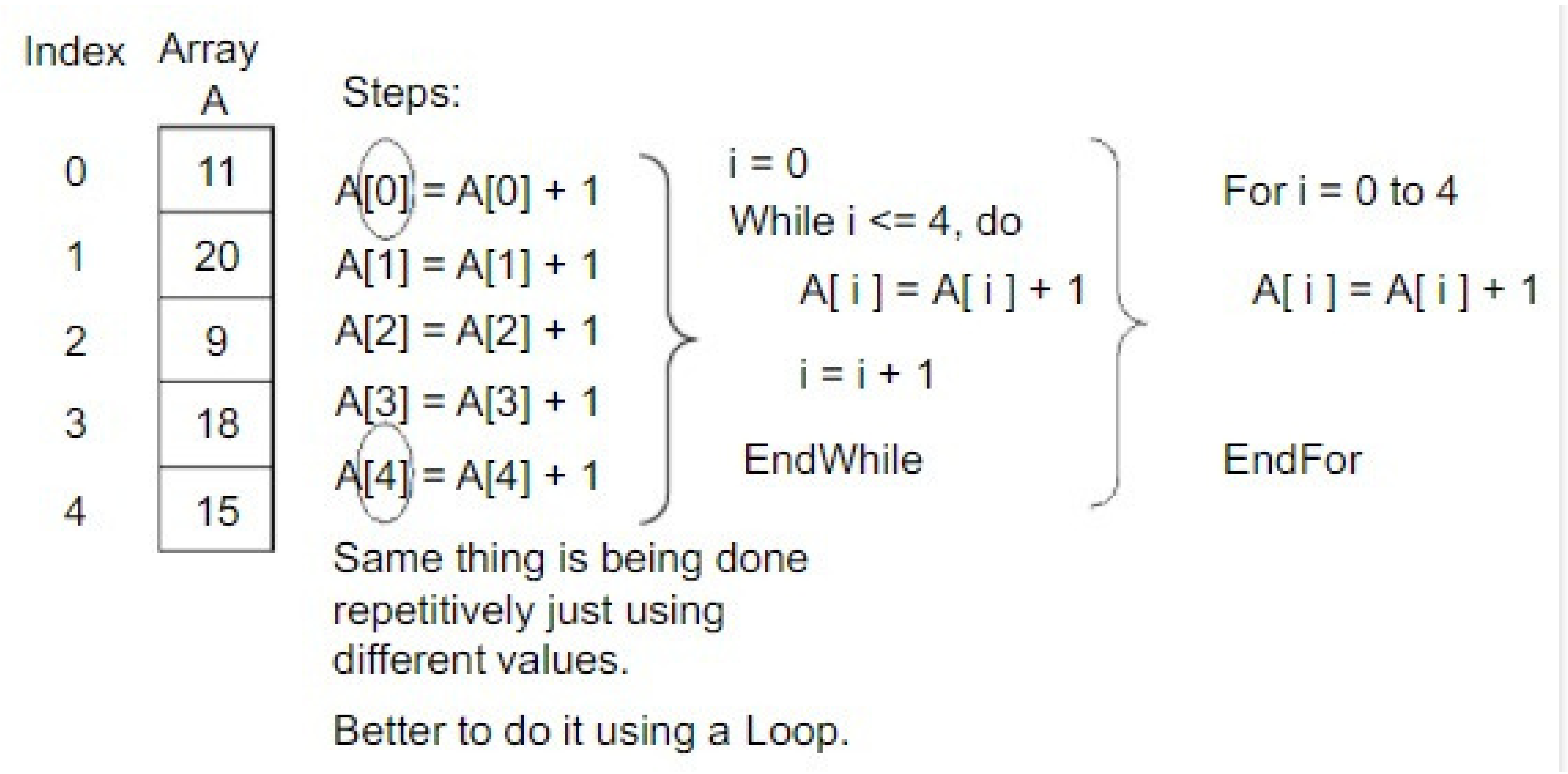
- Manual Method(Without using any Library functions):
- package com.DataFlair.Merge;
- class Manualmerge
- {
- voidmain()
- {
- int arr1[] = {1,2,3,4,5};
- int arr2[] = {6,7,8,9};
- int arr1L = arr1.length;
- int arr2L =arr2.length;
- int arr3L = arr1L + arr2L;

int[] arr3 =newint[arr3L]; for(int i= 0; i< arr1L; i= i+ 1) { arr3[i] = arr1[i]; } for(int i = 0; i < arr2L; i = i + 1) { arr3[arr1L + i] = arr2[i]; } for(int i = 0; i < arr3L; i =i + 1) { System.out.print(arr3[i]); } } }

Question 5

- write a pseudocode algorithm to Increment all the array values by 1

Answer question 5 :



Index	Array A
0	11
1	20
2	9
3	18
4	15

Steps:

```
A[0] = A[0] + 1
A[1] = A[1] + 1
A[2] = A[2] + 1
A[3] = A[3] + 1
A[4] = A[4] + 1
```

```
i = 0
While i <= 4, do
    A[ i ] = A[ i ] + 1
    i = i + 1
EndWhile
```

```
For i = 0 to 4
    A[ i ] = A[ i ] + 1
EndFor
```

Same thing is being done repetitively just using different values.

Better to do it using a Loop.

Question 6

Data[] is an array that is declared as intData[20]; and contains the following values:

Data[] = {12, 23, 34, 45, 56, 67, 78, 89, 90, 100};

a) Calculate the length of the array.

(b) Find the upper_boundand lower_bound.

(c) Show the memory representation of the array.

(d) If a new data element with the value 75 has to be inserted, find its position.

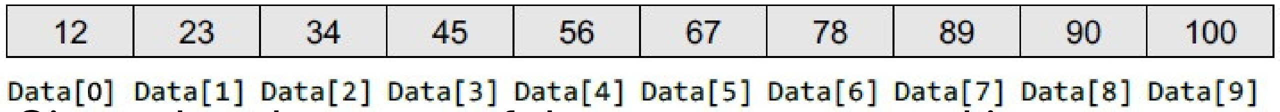
(e) Insert a new data element 75 and show the memory representation after the insertion.

Answer – Question 6

(a) Length of the array = number of elements,
Therefore, length of the array = 10

(b) By default, lower_bound= 0 and upper_bound= 9

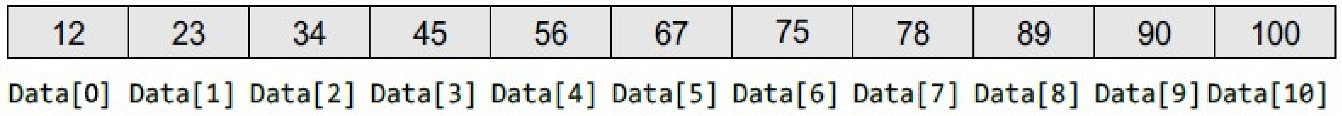
(c)



Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
12	23	34	45	56	67	78	89	90	100

(d) Since the elements of the array are stored in
(d)ascending order, the new data element will be stored after 67, i.e., at the 6th location. So, all the array elements from the 6thposition will be moved one position towards the right to accommodate the new value.

(e)



Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]	Data[10]
12	23	34	45	56	67	75	78	89	90	100

Question 7

- Write an algorithm, pseudo code, and a C program to merge two sorted arrays into a single sorted array.
- Requirements:
  - Accept the sizeand elementsof twoarraysfromthe user.
  - Assume both arrays are already sorted in ascending order.
  - Mergethemintoanewarraysuchthatthefinalarrayisalso sorted in ascending order.
  - Display the merged sorted array.
- Example

Array 1-	20	30	40	50	60							
Array 2-	15	22	31	45	56	62	78					
Array 3-	15	20	22	30	31	40	45	50	56	60	62	78

Answer – Question 7

```
Step 1: Start
Step 2: Input size n1 and elements of the first sorted array A[]
Step 3: Input size n2 and elements of the second sorted array B[]
Step 4: Create a new array C[] to store the merged result
Step 5: Initialize three variables:
        i= 0 (index for A), j = 0 (index for B), k = 0 (index for C)
Step 6: Repeat while both arrays have elements left:
        If A[i] < B[j], then
                C[k] = A[i], i= i+ 1
        Else C[k] = B[j], j = j + 1

        k= k + 1
Step 7: Copy any remaining elements of A[] into C[]
Step 8: Copy any remaining elements of B[] into C[]
Step 9: Display array C[] as the merged sorted array
Step 10: Stop
```

Answer – Question 7

```
BEGIN
   INPUT n1
   INPUT array A[1..n1]
   INPUT n2
   INPUT array B[1..n2]
   i← 0, j ← 0, k ← 0
   WHILE i< n1  AND j < n2 DO
      IF A[i] < B[j] THEN
         C[k] ← A[i]
         i← i+ 1
      ELSE
         C[k] ← B[j]
         j ← j + 1
      ENDIF
      k ← k + 1
   ENDWHILE
   WHILE i< n1 DO
      C[k] ← A[i]
      i← i+ 1
      k ← k + 1
   ENDWHILE

   WHILE j < n2 DO
      C[k] ← B[j]
      j ← j + 1
      k ← k + 1
   ENDWHILE

   DISPLAY "Merged Sorted Array:", C
END
```

Answer –Question 7

```
#include <stdio.h>
int main() {
   int n1, n2, i, j, k;
   int A[50], B[50], C[100];
   // Input first sorted array
   printf("Enter number of elements in first array: ");
   scanf("%d", &n1);
   printf("Enter %d sorted elements: ", n1);
   for(i= 0; i< n1; i++)
      scanf("%d", &A[i]);
// Input second sorted array
   printf("Enter number of elements in second
array: ");
   scanf("%d", &n2);
   printf("Enter %d sorted elements: ", n2);
   for(j = 0; j < n2; j++)
      scanf("%d", &B[j]);
   i= 0; j = 0; k = 0;
   // Merge the two arrays
   while(i< n1 && j < n2) {
      if(A[i] < B[j])
         C[k++] = A[i++];
      else
         C[k++] = B[j++];
   }
```

Answer – Question 7

```
// Copy remaining elements
    while(i< n1)
        C[k++] = A[i++];
    while(j < n2)
        C[k++] = B[j++];
    // Display merged sorted array
    printf("Merged Sorted Array: ");
    for(i= 0; i< k; i++)
        printf("%d ", C[i]);
    printf("\n");
    return 0;
}
```

Try the following activities by yourself. Submit a zip file including all the c files and the pdf (IndexNo_Practical01) with the screenshots of your outputs.

Activity 1.1

- Write a program to find the mean of n numbers using arrays.
- Write a program to print the position of the smallest number of n numbers

  using arrays.
- Write a program to find the second largest of n numbers using an array
- Write a program to enter n number of digits. Form a number using these digits.

Activity 1.2

- Write a program to find whether the array of integers contains a duplicate number.

Activity 1.3

- Write a program to insert a number in an array that is already sorted in ascending order

Activity 1.4

- Write a program to delete a number in an array that is already sorted in ascending order

Activity 1.5

- Write a program to merge two arrays
- Extend the above program merge any number of arrays