**Instructions: Please read carefully**

* Please rename this file as only your ID number **(e.g. 18-\*\*\*\*\*-1.doc or 18-\*\*\*\*\*-1.pdf).**
* Submit the file before **11:00am on 09/06/2021** in the Portal Performance section labeled **Lab task 2. If you cannot complete the full task, do not worry. Just upload what you have completed.**

**Do not copy!!!**

|  |
| --- |
| 1. Initialize TWO integer arrays of different sizes. Merge the input arrays and create a new array. Then print the new array in reverse order.   For example,  Array\_1 = **{10,20,30,40,50}**  Array\_2 = **{1,2,3,4,5,6,7,8}**  Output: **8 7 6 5 4 3 2 1 50 40 30 20 10** |
| **Your code here:**  /\*1.Initialize TWO integer arrays of different sizes. Merge the input arrays and  create a new array. Then print the new array in reverse order.\*/  #include<iostream>  using namespace std;  void mergearr(int arr1[], int arr2[], int n1,int n2, int arr3[])  {  int i = 0, j = 0, k = 0;  while (i<n1 && j <n2)  {  if (arr1[i] < arr2[j])  arr3[k++] = arr1[i++];  else  arr3[k++] = arr2[j++];  }  while (i < n1)  arr3[k++] = arr1[i++];  while (j < n2)  arr3[k++] = arr2[j++];  }  int main()  {  int arr1[5];  cout<<"enter value of arr1"<<endl;  for(int i=0;i<=4;i++)  {  cin>>arr1[i];  }  int n1 = sizeof(arr1)/sizeof(arr1[0]);  int arr2[7];  cout<<"enter value of arr2"<<endl;  for(int j=0;j<=7;j++)  {  cin>>arr2[j];  }  int n2 = sizeof(arr2)/sizeof(arr2[0]);  int arr3[n1+n2];  mergearr(arr1, arr2, n1, n2, arr3);  cout << "merge array" <<endl;  for (int k=0; k < n1+n2; k++)  cout << arr3[k] << " ";  cout<<'\n'<<"reverse array of arr1"<<endl;  for(int i=4;i>=0;i--)  {  cout<<arr1[i]<<" ";  }  cout<<'\n'<<"reverse array of arr2"<<endl;  for(int j=7;j>=0;j--)  {  cout<<arr2[j]<<" ";  }  cout<<'\n'<<"reserve array"<<endl;  for(int k=12;k>=0;k--)  {  cout<<arr3[k]<<" ";  }  cout<<'\n'<<"reserve array"<<endl;  for(int i=12;i>=0;i--)  {  cout<<arr3[i]<<" ";  }  return 0;  } |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize TWO integer arrays **A** and **B** of different sizes. Make a new array with the common elements between **A** and **B**. Print the new array element(s). If there is no common element, output “No common element!”.   For example,  Scenario 1:  Array\_1 = **{1,4,6,3,6,9}**  Array\_2 = **{5,3,7,1,2,6}**  Output: **1 6 3**  Scenario 2:  Array\_1 = **{1,4,6,3,6,9}**  Array\_2 = **{5,8,7,12,21,63}**  Output: **No common element!** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize an array. Size should be more than FIVE. Write your program to change the array in such a way so that there cannot be any duplicate element in the array anymore. Print the changed array. If the initialized array already had no duplicate elements from the beginning, output a message saying “Array already unique!”;   For example,  Scenario 1:  Array\_1 = **{1,4,6,3,6,9,1}**  Output: **1 4 6 3 9**  Scenario 2:  Array\_1 = **{1,4,5,3,6,9}**  Output: **Array already unique!** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize an integer array **A** of size 10. Take an integer as input and print how many times that integer occurs in **A**.   For example,  Array\_1 = **{8,4,6,1,6,9,6,1,9,8}**  Output:  **Input a number to search: 6**  **The number occurs 3 times in the array** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize an integer array of size 10. Print the number of time each element occurs in the array.   For example,  Array\_1 = **{8,4,6,1,6,9,6,1,9,8}**  Output:  **8 occurs = 2 times**  **4 occurs = 1 time**  **6 occurs = 3 times**  **1 occurs = 2 times**  **9 occurs = 2 times** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize a matrix of 3x3 (row x column) size multiply with another matrix of 3x3. Output its multiplied matrix.   For example,  Matrix\_1:  **1 6 7**  **2 4 8**  **3 1 9**  Matrix\_2:  **1 6 7**  **2 4 8**  **3 1 9**  Output:  **34 37 118**  **34 36 118**  **32 31 110** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |

|  |
| --- |
| 1. Initialize a matrix of minimum 3x4 (row x column) size. Output its transpose matrix.   For example,  Matrix\_1:  **1 6 7 9**  **2 4 8 5**  **3 1 9 4**  Output:  **1 2 3**  **6 4 1**  **7 8 9**  **9 5 4** |
| **Your code here:** |
| **Your whole Screenshot here: (Console Output):** |