**Lab Journal 1**

Name: Saad Ahmad

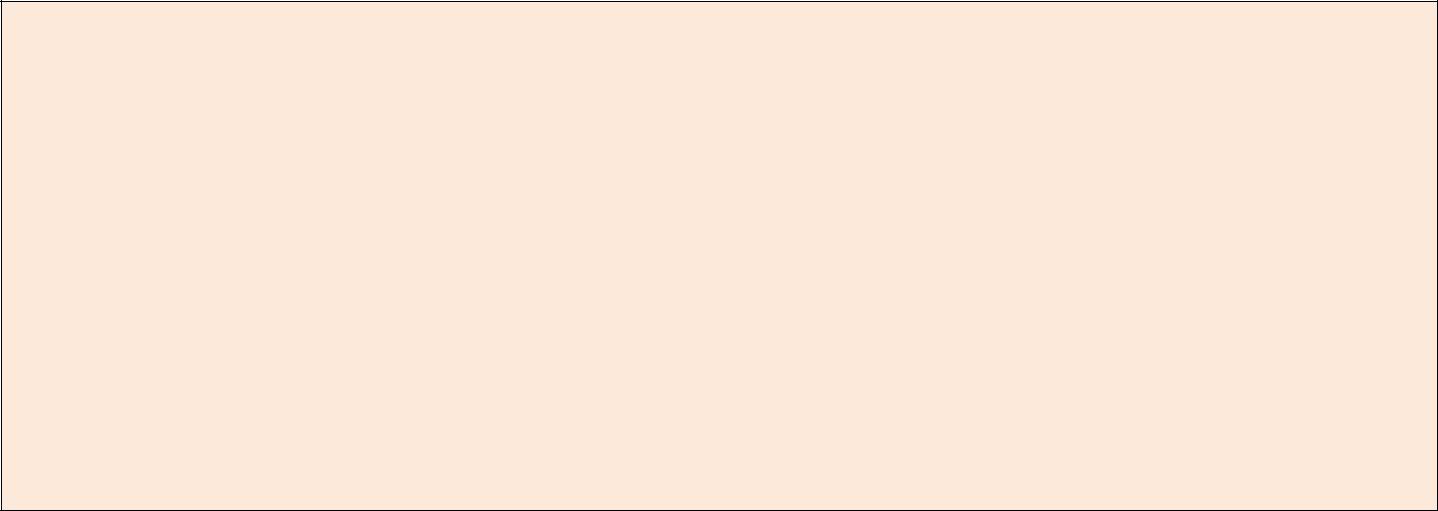
Enrollment #: 01-134222-130

Class/Section: BS-CS-3A

**Task 1: Give answers to the following.**

|  |
| --- |
| 1. Write the prototype of a function named fnct() which accepts an array of integers, a pointer to double, a float by reference and returns a character.   char func(int arr[], double\* val, float& val2); |
| 1. Print the elements of the array ‘a’ using the mentioned notations:     #include <iostream>  using namespace std;  int main() {  int a[] = { 1,2,3,4,5 };  int\* p;  p = a;  cout << "Subscript Notation with array" << endl;  for (int i = 0; i < 5; i++) {  cout << "a[" << i << "] = " << a[i] << endl;  }  cout << "Subscript Notation with pointer" << endl;  for (int i = 0; i < 5; i++) {  cout << "a[" << i << "] = " << p[i] << endl;  }  cout << "Offset Notation with array" << endl;  for (int i = 0; i < 5; i++) {  cout << "(a + " << i << ") = " << \*(a+i) << endl;  }  p = a;  cout << "Offset Notation with pointer" << endl;  for (int i = 0; i < 5; i++) {  cout << "(a + " << i << ") = " << \*(p+i) << endl;  }  } |
| 1. What is the difference between function overloading and function overriding?   Ans: In function overloading the parameters of the functions are different while the name is same.  In function overriding the name of the function is same but the compiler decides which one to use at runtime. |
| 1. Write C++ statement(s) to allocate space for 10 doubles (using dynamic memory allocation).   double\* val = new double[10] ; |
| 1. Study the given program and determine what the program is intended to do. (Hint: Dry run the program with array {1,2,1,2,3} and analyze the output).     Answer: 1 2 3 |

**Task 2: Implement the given exercises.**

**Exercise 1:**

Let A and B be two sets: (Data of sets can be integer, double or float)  
A = {1.3, 23.21, 3.3, 65.09, 5.32, 89.1, 101.20, 43.4}  
B = {7.17, 43.4, 9.91, 87.73, 11.00, 43.21, 15.34, 99.54, 23.21, 1.3, 5.32}

**Write the following functions:**• **Print**Prints all the elements of the array.  
• **Union**Find the union of A and B. The union of A and B is the set that contains those  
elements that are either in A or in B, or in both c)  
• **Remove duplicate**Remove the duplicates in the resultant set.  
• **Delete**Deletes specified element from the sets.  
• **Set size**Set the size of the set.  
• **Intersection**Find the intersection of A and B. The intersection of A and B is the set that  
contains those elements that are in both A and B.

#include <iostream>

using namespace std;

int main() {

float A[] = { 1.3, 23.21, 3.3, 65.09, 5.32, 89.1, 101.20, 43.4 };

float B[] = { 7.17, 43.4, 9.91, 87.73, 11.00, 43.21, 15.34, 99.54, 23.21, 1.3, 5.32 };

int sizeA = sizeof(A) / sizeof(float);

int sizeB = sizeof(B) / sizeof(float);

cout << "Array A: " << endl;

for (int i = 0; i < sizeA; i++) {

cout << A[i] << ", ";

}

cout << endl;

cout << "Array B: " << endl;

for (int i = 0; i < sizeB; i++) {

cout << B[i] << ", ";

}

int size = sizeA + sizeB;

float\* C = new float[size];

int k;

for (k = 0; k < sizeA; k++) {

C[k] = A[k];

}

for (int j = 0; k < size; k++) {

C[k] = B[j];

j++;

}

cout << endl;

cout << "Array C: " << endl;

for (int i = 0; i < size; i++) {

cout << C[i] << ", ";

}

for (int i = 0; i < size; i++) {

for (int j = i + 1; j < size; j++) {

if (C[i] == C[j]) {

C[j] = 0;

}

}

}

cout << endl;

cout << "Array C with no duplicates: " << endl;

for (int i = 0; i < size; i++) {

if (C[i] != 0) {

cout << C[i] << ", ";

}

}

float check;

cout << endl;

cout << "Enter the number you want to delete: " << endl;

cin >> check;

for (int i = 0; i < size; i++) {

if (C[i] == check) {

C[i] = 0;

}

}

cout << "Array C: " << endl;

for (int i = 0; i < size; i++) {

if (C[i] != 0) {

cout << C[i] << ", ";

}

}

int sizeD;

cout << endl;

cout << "Enter the size of the array: " << endl;

cin >> sizeD;

float\* D = new float[sizeD];

for (int i = 0; i < sizeD; i++) {

D[i] = C[i];

}

cout << "Array D: " << endl;

for (int i = 0; i < sizeD; i++) {

cout << D[i] << ", ";

}

int counter = 0;

for (int i = 0; i < sizeA; i++) {

for (int j = 0; j < sizeB; j++) {

if (A[i] == B[j]) {

counter++;

}

}

}

cout << endl;

cout << "Common elements between A and B: " << counter << endl;

int count = 0;

float\* E = new float[counter];

for (int i = 0; i < sizeA; i++) {

for (int j = 0; j < sizeB; j++) {

if (A[i] == B[j]) {

E[count] = A[i];

count++;

}

}

}

cout << "Array E: " << endl;

for (int i = 0; i < counter; i++) {

cout << E[i] << ", ";

}

delete[] C;

delete[] D;

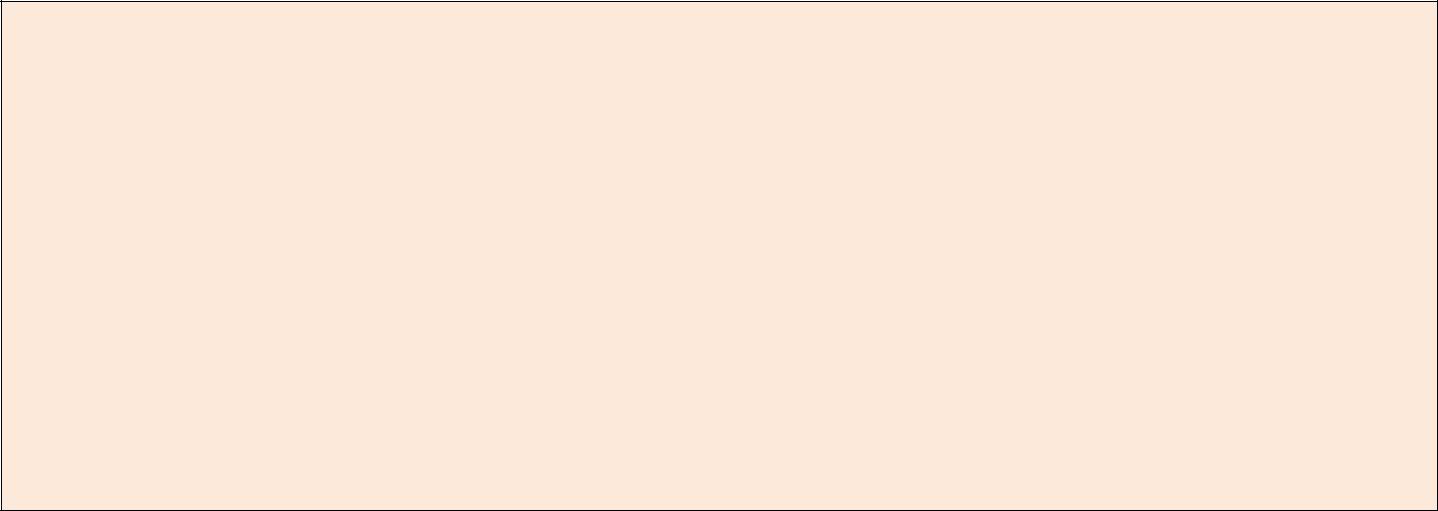
delete[] E;

return 0;

}

A screenshot of a computer program

Description automatically generated

**Exercise 2:**

Write a program with a function which accepts an array of integers and a key value. The function should return the sum of all the multiples of the key value in the array. For example, for the array {1, 4, 10, 12, 15, 20, 22} and the key value 5, the function should return the sum 10+15+20.

#include <iostream>

using namespace std;

int func(int\* arr,int size, int keyvalue) {

int sum = 0;

for (int i = 0; i < size; i++) {

if (arr[i] % keyvalue == 0) {

sum = sum + arr[i];

}

}

return sum;

}

int main() {

int keyval;

int arr[] = {1,4,10,12,15,20,22};

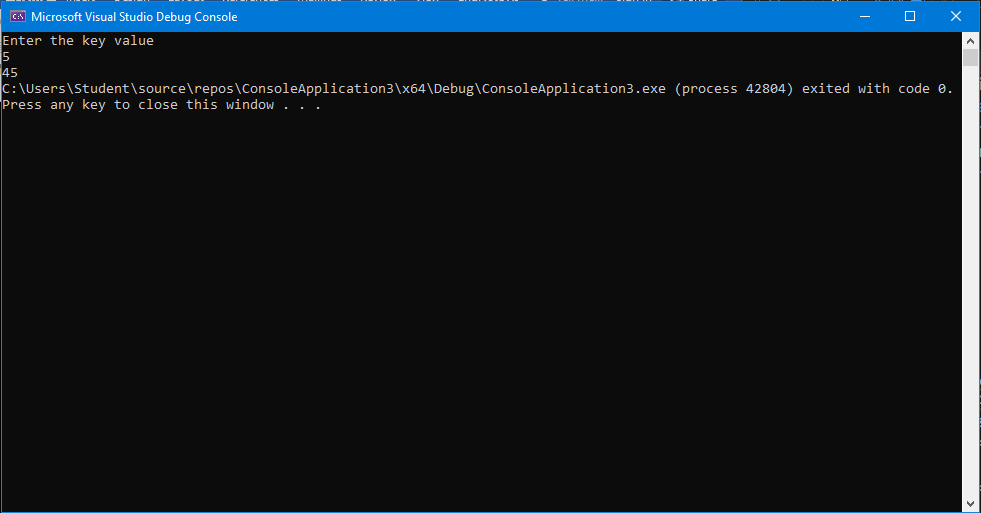
int size = sizeof(arr) / sizeof(int);

cout << "Enter the key value " << endl;

cin >> keyval;

cout << func(arr, size, keyval);

}

****

**Implement the given exercises and get them checked by the instructor. If you are unable to complete the tasks in the lab session, deposit this journal along with your programs (printed or handwritten) before the start of the next lab session.**

|  |  |
| --- | --- |
| **Submitted to:** |  |
| **Task 1:** |  |
| **Task 2:** |  |
| **Date:** |  |
| **Total marks:** |  |