**PF LAB-LAB 10**

**Name: Ahmed Kasteer**

**Section: 1C**

**Question 1:**

#include<iostream>

#include <ctime>

using namespace std;

int main()

{

int largest = -99999999;

int array[5][5];

srand(time(0));

for (int i = 0; i <=4 ; i++)

{

for (int j = 0; j <= 4; j++)

{

array[i][j] = rand() % 50 + 1;

cout << array[i][j] << " ";

if( array[i][j] > largest)

{

largest = array [i][j];

}

}

cout << endl;

}

cout << "Largest number from above array is:" << endl;

cout << largest << endl;

return 0;

}

Text

Description automatically generated

**Question 2:**

#include <iostream>

using namespace std;

void twoD\_enter(int arr[][3])

{

cout << "Enter Array for 2D Elements."<< endl;

for (int i = 0; i <=2; i++)

{

for (int j = 0; j <=2; j++)

{

cin >> arr[i][j];

}

}

}

void conversion(int arr1[9], int arr2[][3])

{

for (int i= 0; i < 9; i++)

{

for (int j = 0; j < 3 ; j ++)

{

arr1[i \* 3 + j] = arr2[i][j];

}

}

}

void display1D(int arr1[9])

{

for (int i= 0 ; i < 9; i++)

{

cout << arr1[i] << " ";

}

cout << endl;

}

int main()

{

int array2D[3][3] = {0};

int array1D[9] = {0};

twoD\_enter(array2D);

conversion(array1D, array2D);

display1D(array1D);

return 0;

}

**Text

Description automatically generated with medium confidence**

**Question 3:**

#include<iostream>

#include <fstream>

#include<string>

using namespace std;

void binarySearch(float a[10])

{

int first = 0, last = 9; int middle = 0;

float n;

cout << "Enter the number to found\n";

cin >> n;

while(first <= last)

{

if(a[middle]<n)

first = middle+1;

else if(a[middle]==n)

{

cout<<"The number, "<<n<<" found at Position "<<middle+1;

break;

}

else

last = middle-1;

middle = (first+last)/2;

}

if(first>last)

cout<<"The number, "<<n<<" is not found in given Array";

cout<<endl;

}

int main()

{

float arr[10] = { 0.0 };

int i = 0;

ofstream file("input.txt", ios::out);

file << 1.0 << endl;

file << 2.0 << endl;

file << 3.0 << endl;

file << 4.0 << endl;

file << 5.0 << endl;

file << 6.0 << endl;

file << 7.0 << endl;

file << 8.0 << endl;

file << 9.0 << endl;

file << 10.0 << endl;

file.close();

ifstream readfile("input.txt", ios::in);

while (i < 10)

{

readfile >> arr[i];

i++;

}

readfile.close();

binarySearch(arr);

return 0;

}

**Text

Description automatically generated**