Q.1a:

#include <iostream>

using namespace std;

void printUnique(int arr[])

{

int unique[100], endpointer = 0;

for (int i = 0; i < 100; i++)

{

bool found = false;

for (int j = 0; j < 100; j++)

{

if (arr[i] == arr[j] && i != j)

{

found = true;

break;

}

}

if (!found)

{

unique[endpointer] = arr[i];

endpointer++;

}

}

cout << "Unique elements in the array are: ";

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

cout << unique[i] << ' ';

cout << endl;

}

int main()

{

int array[100], input;

for (int i = 0; i < 100; i++)

{

cout << "Enter number to be put in array(Enter -1 to exit):";

cin >> input;

if (input == -1)

break;

array[i] = input;

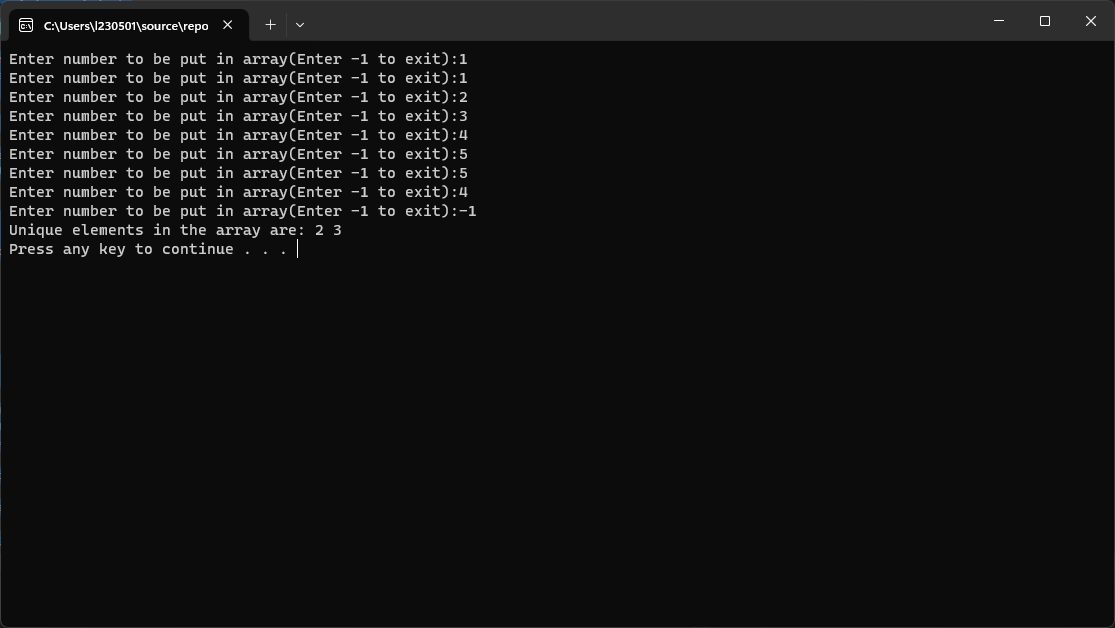
}

printUnique(array);

system("pause");

return 0;

}



Q.1b:

#include <iostream>

using namespace std;

void cycleRotate(int arr[], int d)

{

if (d > 0)

{

int temp;

temp = arr[0];

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

arr[i] = arr[i + 1];

arr[9] = temp;

cycleRotate(arr, d - 1);

}

else if (d < 0)

{

int temp;

temp = arr[9];

for (int i = 9; i > 0; i--)

arr[i] = arr[i-1];

arr[0] = temp;

cycleRotate(arr, d + 1);

}

}

int main()

{

int array[10], input, d;

for (int i = 0; i < 10; i++)

{

cout << "Enter number to be put in array holding 10 integers:";

cin >> input;

array[i] = input;

}

cout << "Enter factor d to rotate left by:";

cin >> d;

cycleRotate(array, d);

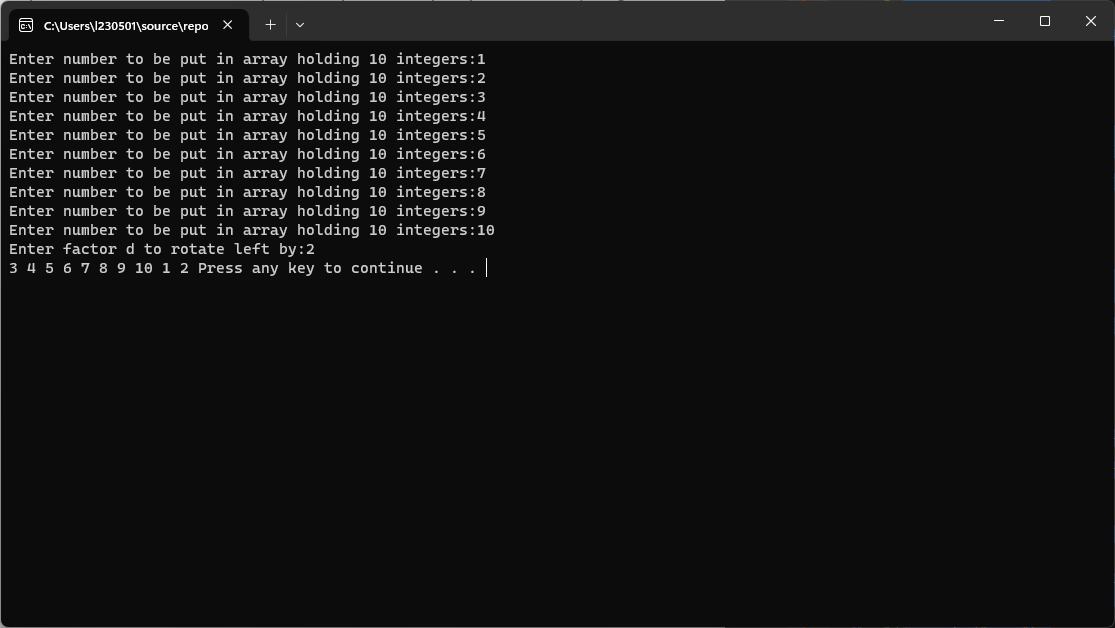
for (int i = 0; i < 10; i++)

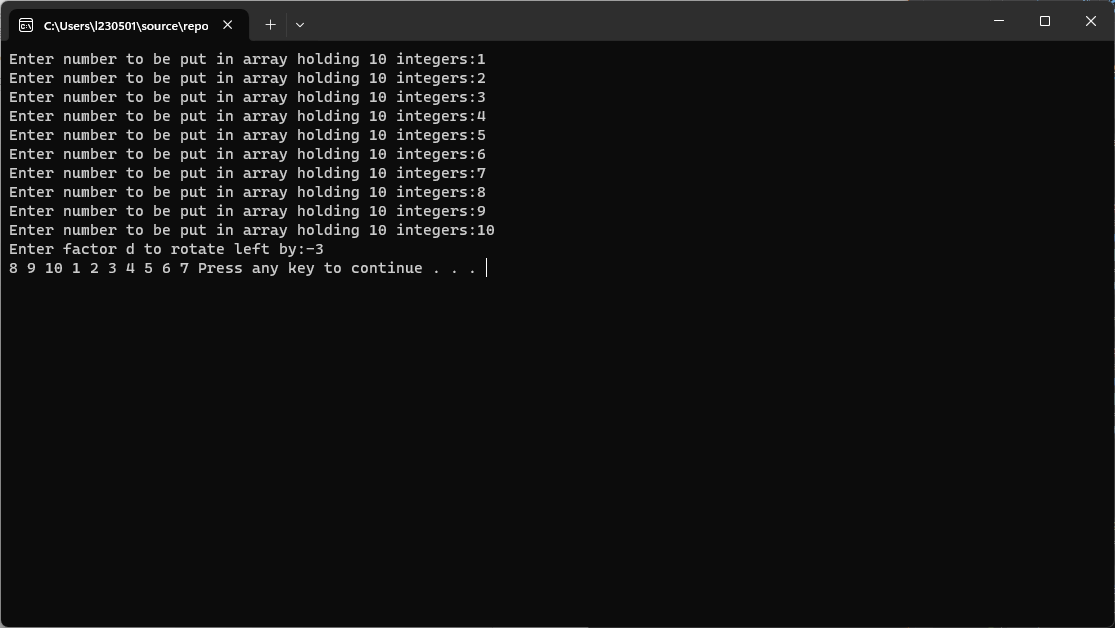
cout << array[i] << ' ';

system("pause");

return 0;

}





Q.2a:

#include <iostream>

using namespace std;

bool linearSearch(int arr[], int value)

{

for (int i = 0; i < 10; i++)

{

if (arr[i] == value)

return true;

}

return false;

}

int main()

{

int array[10] = {13579, 26791, 26792, 33445, 55555, 62483, 77777, 79422, 85647, 93121};

int ticket;

do

{

cout << "Enter a valid 5-digit ticket number: ";

cin >> ticket;

if (ticket > 9999 && ticket < 100000)

break;

} while (true);

if (linearSearch(array, ticket) == true)

cout << "Congratulations, you have won!!!\n";

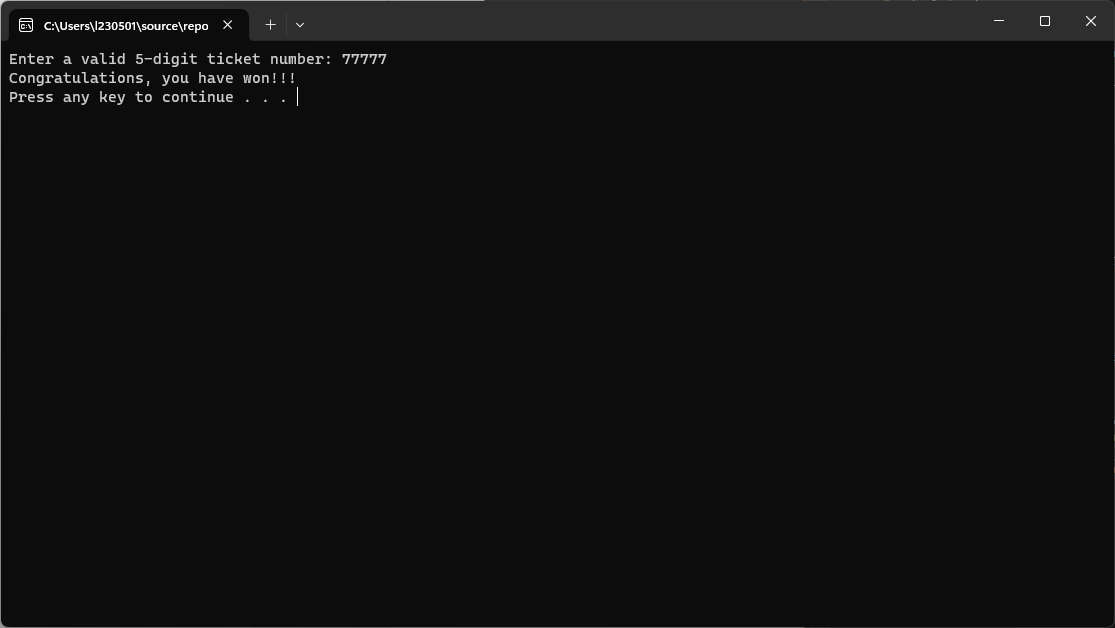
else

cout << "You are not among the winners.\n";

system("pause");

return 0;

}



Q.2b:

#include <iostream>

using namespace std;

bool binarySearch(int arr[], int value)

{

int start = 0, end = 9, mid;

while (start <= end)

{

mid = (start + end) / 2;

if (arr[mid] == value)

return true;

else if (value > arr[mid])

start = mid + 1;

else

end = mid - 1;

}

return false;

}

int main()

{

int array[10] = {13579, 26791, 26792, 33445, 55555, 62483, 77777, 79422, 85647, 93121};

int ticket;

do

{

cout << "Enter a valid 5-digit ticket number: ";

cin >> ticket;

if (ticket > 9999 && ticket < 100000)

break;

} while (true);

if (binarySearch(array, ticket) == true)

cout << "Congratulations, you have won!!!\n";

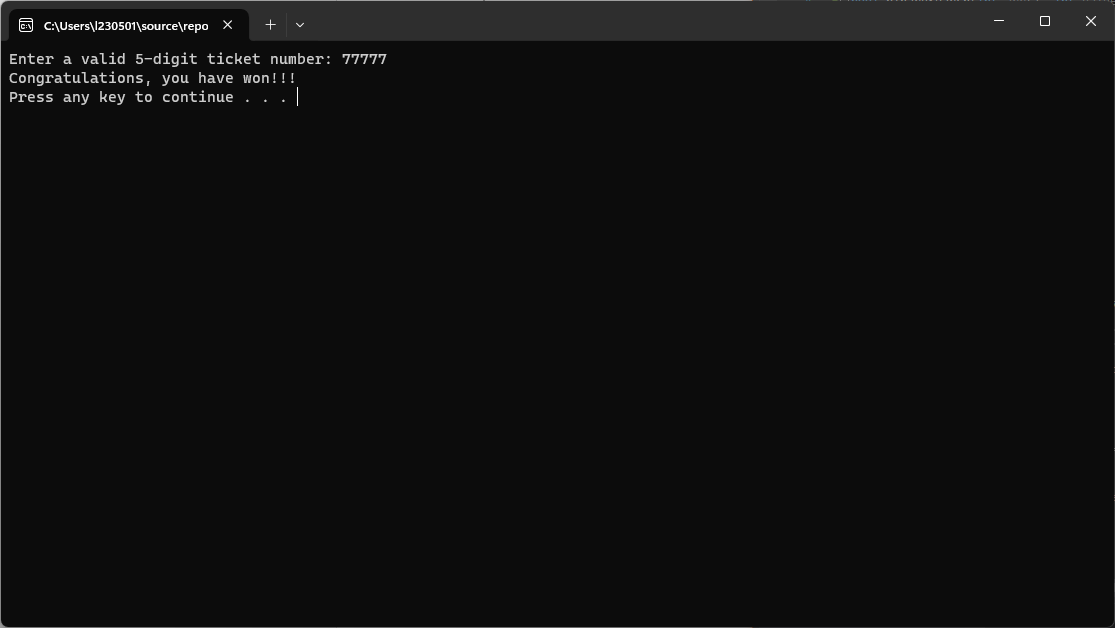
else

cout << "You are not among the winners.\n";

system("pause");

return 0;

}



Q.3:

#include <iostream>

using namespace std;

int countinsertion = 0, countselection = 0;

void insertionSort(int arr[])

{

int key, j;

for (int i = 1; i < 20; i++) {

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

countinsertion++;

}

arr[j + 1] = key;

}

}

void selectionSort(int arr[])

{

for (int i = 0; i < 19; i++) {

int min = i;

for (int j = i+1; j < 20; j++) {

if (arr[j] < arr[min])

min = j;

}

int temp = arr[min];

arr[min] = arr[i];

arr[i] = temp;

countselection++;

}

}

int main()

{

int array1[20], array2[20], input;

for (int i = 0; i < 20; i++)

{

cout << "Enter number to be put in array: ";

cin >> input;

array1[i] = input;

}

for (int i = 0; i < 20; i++)

array2[i] = array1[i];

insertionSort(array1);

selectionSort(array2);

cout << "Count of swaps in insertion sort is: " << countinsertion << endl;

cout << "Count of swaps in selection sort is: " << countselection << endl;

cout << "Array sorted by insertion sort:\n";

for (int i = 0; i < 20; i++)

cout << array1[i] << ' ';

cout << "\nArray sorted by selection sort:\n";

for (int i = 0; i < 20; i++)

cout << array2[i] << ' ';

cout << endl;

system("pause");

return 0;

}

