Task 4

#include<iostream>

using namespace std;

int main() {

double balance[5] = { 1000.0,2.01,3.4,17.0,50.40 };

double \*p;

p = balance;

cout << "the array is equal to = " << endl; // checking array results

for (int i = 0;i < 5;i++)

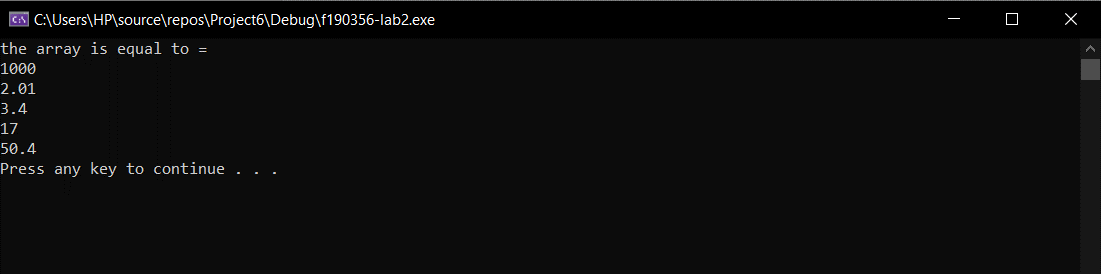
{

cout << \*(p + i) << endl; // array values stored at indexs

}

system("pause");

return 0;

}

**Task 5:**

#include<iostream>

using namespace std;

int main() {

string a;

a = "hello world"; // random string

int b = a.length(); // to check the size of array which will be held in b

char arham[100];

arham[b];

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

{

arham[i] = a[i];

}

//for (int i = b-1;i >= 0;i--)

//{

//

// cout << arham[i] << endl;

//

//}

char\* ptr;

ptr = arham; // for printing the the array backwards

for (int i = b - 1;i >= 0;i--)

{

cout << \*(ptr + i); // value of array (backwards and index by index)

}

// program by Arham xD

system("pause");

return 0;

}



**Task 6:**

#include<iostream>

using namespace std;

int arrray[5] = { 3,9,1,15,3 };

int\* ptr;

int main() {

ptr = arrray;

{

for (int i = 0;i < 5;i++)

{

cout << \*ptr << endl;

cout << ptr << endl;

ptr++; // increment in ptr

}

}

cout << "-----------------";

{

for (int i = 0;i < 6;i++)

{

cout << \*ptr << endl;

cout << ptr << endl;

ptr--; // decrement in array

}

}

cout << "ptr is equal at = "; // point where both arrays are equal

{

for (int i = 0;i < 6;i++)

{

if (\*ptr = \*ptr)

{

cout << \*ptr << endl;

break;

}

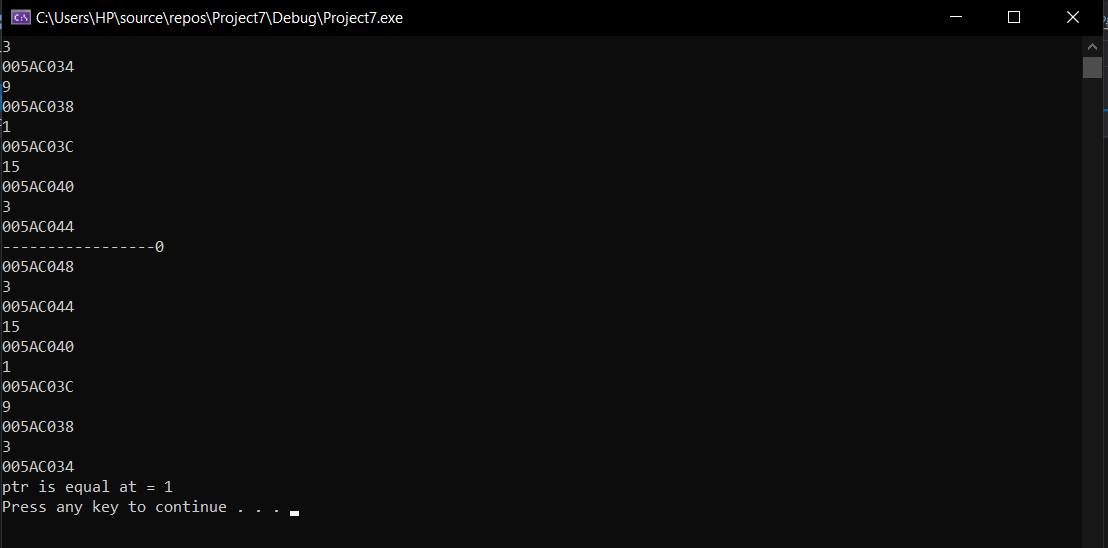
}

}

// program by Arham xD

system("pause");

return 0;

}

**Task 7:**

#include<iostream>

using namespace std;

void primaryCheck(int \*ptr)

{

int b;

b = \*ptr;

int counter = 0;

for (int i = 2; i <= b / 2; i++) {

if (b % i == 0)

{

counter++;

}

}

if (counter == 0)

{

cout << "this is a prime number ! " << endl;

}

else

cout << "NO, this is not a prime number ! " << endl;

}

int main() {

cout << "please enter the number which you want to check = " << endl;

int a;

int\* ptr;

cin >> a;

ptr = &a;

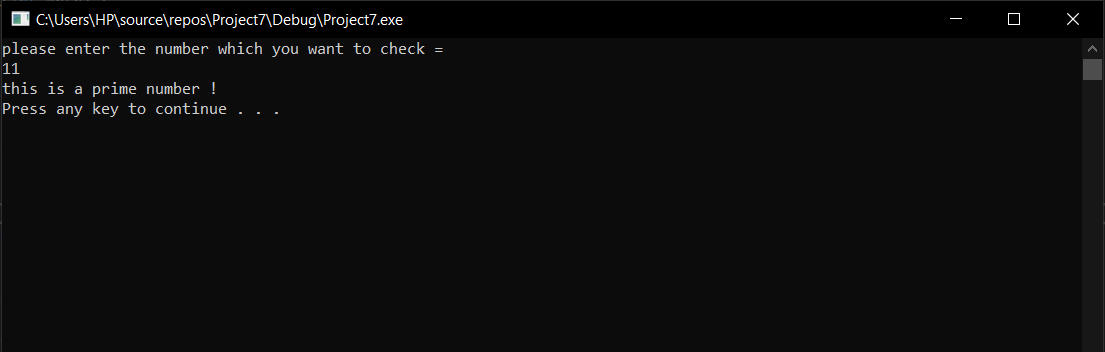
// program by Arham xD

primaryCheck(ptr);

system("pause");

return 0;

}



**Task 8:**

#include<iostream>

using namespace std;

void matrix(int\*\* ptr) {

int x;

x = \*\*ptr;

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

{

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

{

cout << x;

}

cout << endl;

}

}

int main() {

int num;

int \*ptr1;

cout << "enter any number = " << endl;

cin >> num;

ptr1 = &num;

int \*\*ptr;

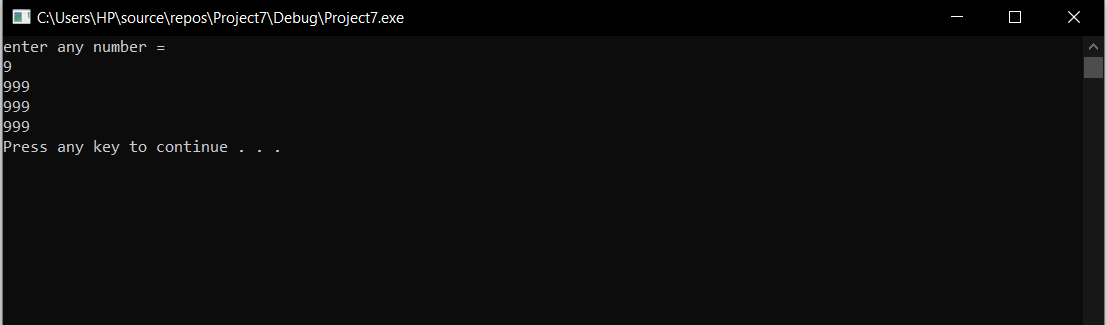
ptr = &ptr1;

matrix(ptr);

// program by Arham xD

system("pause");

return 0;

}

**Task 7:**

#include<iostream>

using namespace std;

int main()

{

int firstnumber;

int secondnumber;

firstnumber = 5;

secondnumber = 15;

int\* p1, \* p2, \*\* p3, \*\* p4;

p1 = &firstnumber; // p1 holds the address of first number

p2 = &secondnumber; // p2 holds the address of second number

p3 = &p1; // double pointer p3 holds address of p1

p4 = &p2; // double pointer p4 holds address of p2

\*p1 = 10; // p1 now holds number 10 in it

\*p2 = \*p1; // both hold the same value

p1 = p2; // copied addresses

cout << "the address of p1 is = " << &p1 << "the address of p2 is = " << &p2 << endl;

p3 = p4; //address copied

cout << "the address of p3 is = " << &p3 << "the address of p4 is = " << &p4 << endl;

\*p1 = 20;

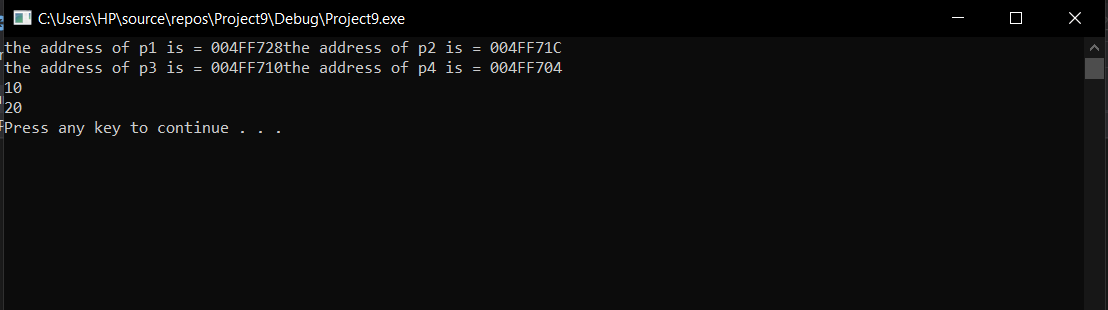
cout << firstnumber<<endl;

cout << secondnumber << endl;

system("pause");

return 0;

}



**Task 1:**

#include<iostream>

#include<fstream>

#include<cstdlib>

#include<time.h>

using namespace std;

void numArray(int numarray[])

{

int c = 0;

int n;

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

{

cout << "enter number = " << endl;

cin >> n;

if (n >= 100 && n <= 999) // limit defined in question

{

numarray[i] = n;

}

else

{

cout << "ENTER VALID NUMBER ONLY = " << endl;

cin >> n;

numarray[i] = n;

}

}

}

void bublesort(int numarray[]) {

numArray(numarray);

int temp;

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

{

for (int j = i + 1;j < 10;j++)

if (numarray[i] > numarray[j])

{

temp = numarray[j];

numarray[j] = numarray[i];

numarray[i] = temp;

}

}

ofstream write;

write.open("num.txt", ios::app);

cout << "BUBLE SORTING successful and written in file called num " << endl;

write << "buble sorting = " << endl;

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

{

write << numarray[i] << endl;

}

}

void selectionsort(int numarray[])

{

int temp, tempo;

numArray(numarray);

int min = 0;

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

{

min = numarray[i];

tempo = i;

for (int j = i + 1;j < 10;j++) // upto size of array

{

if (numarray[j] < min)

{

min = numarray[j];

tempo = j;

}

}

temp = numarray[i]; // sorting and I am using 3 variables for that

numarray[i] = numarray[tempo];

numarray[tempo] = temp;

}

ofstream write;

write.open("num.txt", ios::app); // so that I can write both sorts in one file called num

cout << "SElECTION SORTING successful and written in file called num " << endl;

write << "selection sorting = " << endl;

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

{

write << numarray[i] << endl;

}

write.close();

}

int main() {

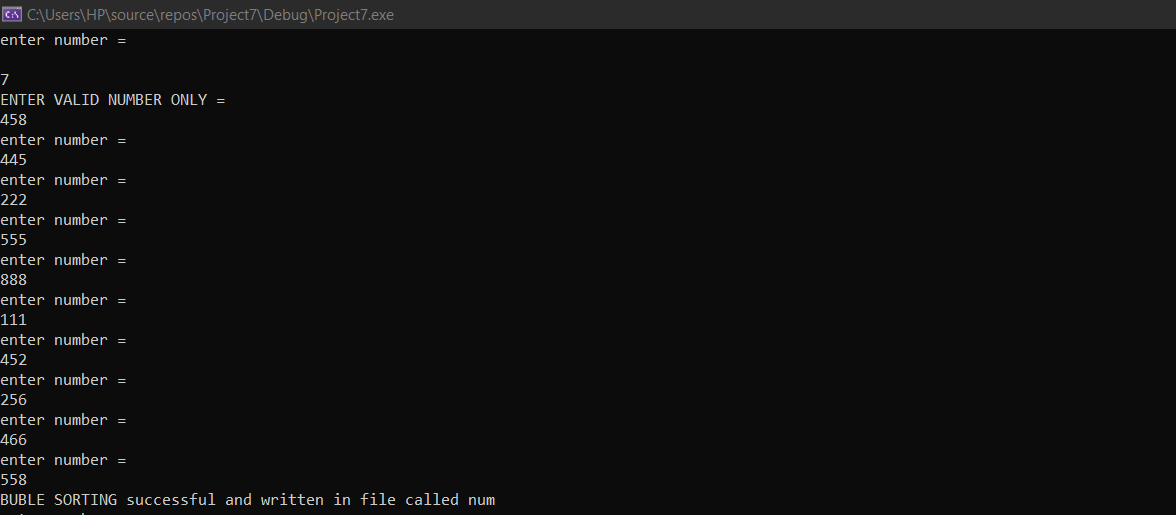
int numarray[10];

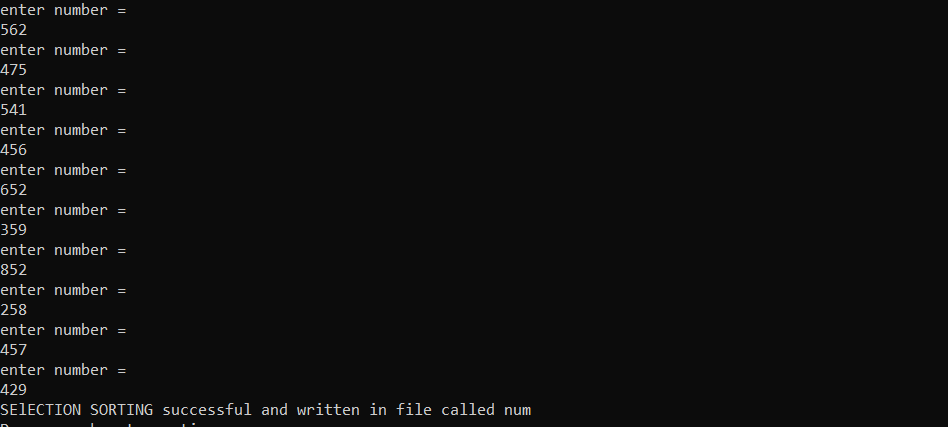
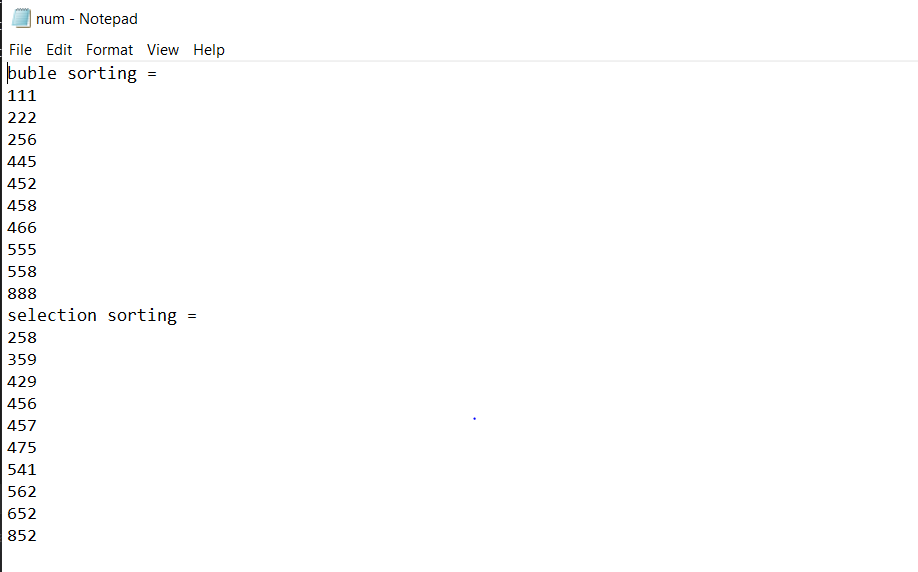
bublesort(numarray); // program by Arham xD

selectionsort(numarray);

system("pause");

return 0;

}



**Task 2**

#include<iostream>

#include<fstream>

#include<cstdlib>

#include<time.h>

using namespace std;

void linearsearch(int array[10]) {

cout << "enter number to find ! = " << endl;

int num;

cin >> num;

int c = 0;

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

{

if (num == array[i]) // array from main function containing selection sort values

{

cout << "NUMBER FOUND AT POSITION/ INDEX = " << i << " AND THE NUMBER IS = " << num << endl; // will print both index and value

}

else c == 1;

}

if (c == 1)

{

cout << "NO SUCH NUMBER WAS TO BE FOUND ! " << endl; // incase an invalid number is entered

}

}

void binarysearch(int array[10])

{

int c = 0;

int v = 0;

cout << "which number do you want to search ! = " << endl;

int num;

cin >> num;

int first, last;

first =0; // this is for assigning the first index

last = 9; // this is for the assignment of last index

int half; //this variable is written so we can find the middle value by which we can perform binary search

while (first <= last)

{

half = (first + last) / 2; // middle value index will be stored later in array , so thats why we are using this variable

if (array[half] == num) // if index value and the value we entered is same or not

{

cout << "ELEMENT FOUND AT POSITION = " << half << "AND THE NUMBER IS = " << num << endl; // in case we find the element

break;

}

else if (array[half] < num)

{

first = half + 1; // for finding the exact value we need

c++;

}

else

{

last = half - 1; // for finding the exact value we need

v++; // the value exits

}

}

if (c == 0 || v == 0)

{

cout << "not found " << endl; //if value doesnt exist

}

}

int main() {

ifstream read; // for reading the data

read.open("num.txt");

string wow;

int readData[10];

{

for (int i = 0;i < 16;i++) // this loop will read the "buble sorting= " line that i ve written in text file

{

{read >> wow;

}

}

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

{

read >> readData[i];

}

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

{

cout << readData[i] << endl;

}

cout << "linear searching = " << endl;

linearsearch(readData); //function called

cout << endl; // program by Arham xD

cout << "BINARY SEARCHING = " << endl; //fucntion called

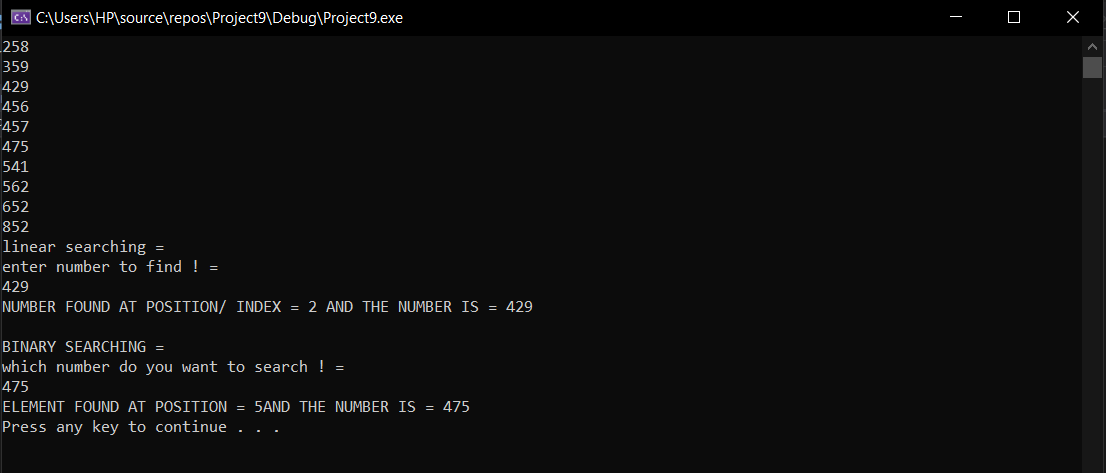
binarysearch(readData);

}

read.close();

system("pause");

return 0;

}

**Task 3:**

#include<iostream>

#include<fstream>

#include<cstdlib>

#include<time.h>

using namespace std;

int main(){

int firstvalue;

int secondvalue;

firstvalue = 5;

secondvalue = 15;

int\* p1, \*p2;

p1 = &firstvalue; // stores the address of firstnumber

p2 = &secondvalue; // stores the address of secondnumber

\*p1 = 10;

cout << "the value pointed by p1 is = " << \*p1<<endl; //printing value of p1

\*p2 = \*p1;

cout << "the value pointed by p2 is = " << \*p2 << endl; //printing value of p2

p1 = p2; // address of values which they hold is copied

\*p1 = 20;

cout << firstvalue << endl; // value should be 10

cout << secondvalue << endl; // value should be 20

system("pause");

return 0;

} 