Task 1:

#include<iostream>

#include<time.h>

using namespace std;

void BubbleSort(int arr[]){

for (int i = 0; i < 10 - 1; i++){

for (int j = 0; j < 10 - i - 1; j++){

if (arr[j] > arr[j + 1]){

swap(arr[j], arr[j + 1]);

}

}

}

cout << "Array with Bubble Sorting is : " << endl;

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

cout << arr[i] << " ";

}

}

int main(){

int arr[10];

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

cin >> arr[i];

while (arr[i] < 100 || arr[i] > 999) {

cout << "Please enter greater than 100 and less than 999" << endl;

cin >> arr[i];

}

}

cout << "Array before Sorting : " << endl;

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

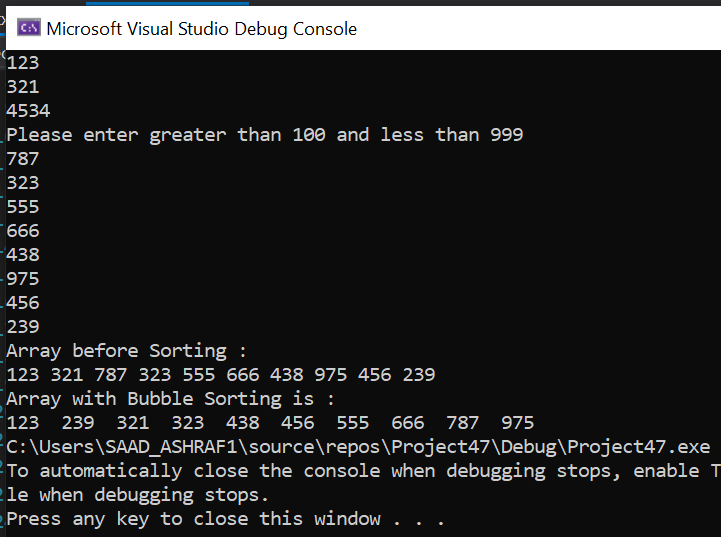
cout << arr[i] << " ";

}

cout << endl;

BubbleSort(arr);

}



Task 2:

#include<iostream>

using namespace std;

void maximum(int a[]) {

int min = a[0];

int c = 0;

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

if (min < a[i]) {

min = a[i];

c = i;

}

}

cout << endl;

cout << "The largest index is : " << c << " and it has value : ";

cout << min;

cout << endl;

}

void linearSearch(int a[]) {

int min = a[0];

int c = 0;

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

if (min > a[i]) {

min = a[i];

c = i;

}

}

cout << "The smallest Index is : " << c << " and it has value : ";

cout << min;

maximum(a);

}

void binarysearch(int arr[]) {

cout << endl;

cout << "After sorting then : " << endl;

for (int i = 0; i < 10 - 1; i++) {

for (int j = 0; j < 10 - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

swap(arr[j], arr[j + 1]);

}

}

}

cout << endl;

cout << "The smallest index is 0 and value is : " << arr[0]<<endl;

cout << "The largest index is 9 and value is : " << arr[9]<<endl;;

}

int main() {

int a[50];

cout << "enter the elements:";

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

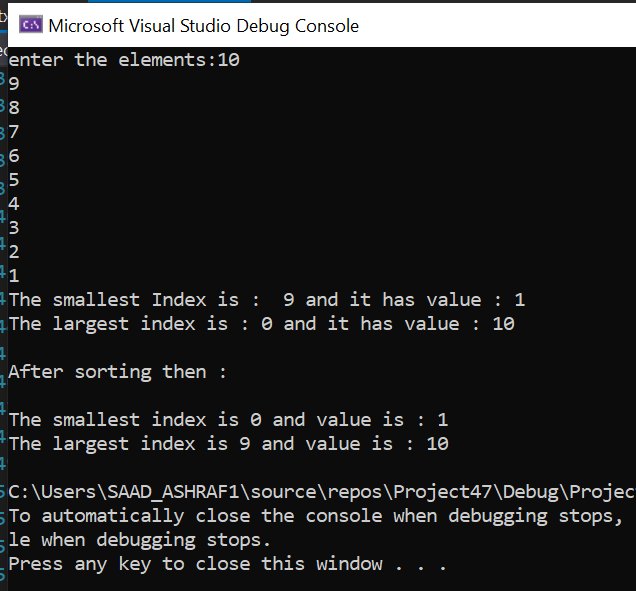
cin >> a[i];

}

linearSearch(a);

binarysearch(a);

}



Task 3:

#include<iostream>

using namespace std;

int main()

{

int firstvalue = 5, secondvalue = 15;

int\* p1, \* p2;

p1 = &firstvalue; ;// p1 = address of firstvalue

p2 = &secondvalue; // p2 = address of secondvalue

\*p1 += 5;

cout << \*p1;// value pointed by p1 = 10

\*p2 = \*p1;

cout << \*p2; // value pointed by p2 = value pointed by p1

p1 = p2; ;// p1 = p2 (address of pointer is copied)

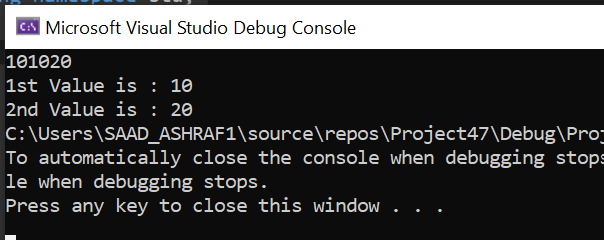
\*p1 += 10;

cout << \*p1 << endl; // value pointed by p1 = 20

cout << "1st Value is : " << firstvalue << endl;// print firstvalue

cout << "2nd Value is : " << secondvalue;// print secondvalue

}



Task 4:

#include<iostream>

using namespace std;

int main(){

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

double\* p; //making a double pointer

p = balance; //pointer is pointing to the array

cout << "Accessing with pointers :" << endl;

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

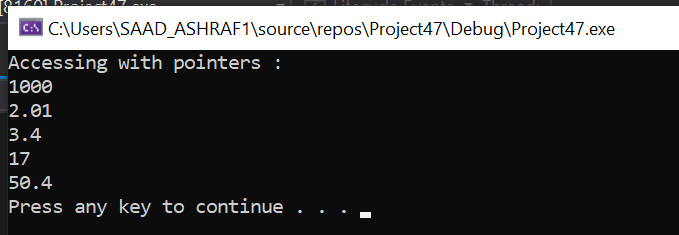
{

cout << \*(p + i) << endl; //logic to point to the next element of array and print it

}

system("pause");

}



Task 5:

#include <iostream>

using namespace std;

void Reverse(char string[]) {

char reverse[50];

char\* ptr = string;

char\* ReversePtr = reverse;

int i = -1;

while (\*ptr)

{

ptr++;

i++;

}

while (i >= 0)

{

ptr--;

--i;

\*ReversePtr = \*ptr;

ReversePtr++;

}

\*ReversePtr =NULL; //initalizig pointer with Null

cout << " Reverse of the string is :" << reverse; //printing

}

int main()

{

char st[50];

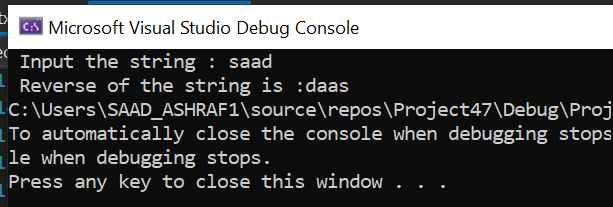
cout << " Input the string : ";

cin >> st;

Reverse(st); //calling function

return 0;

}



Task 6:

#include<iostream>

using namespace std;

int main(){

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

int\* ptr = arr;

cout << "Increment" << endl;

//increment

int i = 0;

do{

cout << " var" << i << " address : "<< ptr <<" "<< "Value of var " << i << " is : "<< \*ptr << endl;

ptr++;

i++;

} while (i < 5);

cout << "deecrement" << endl;

//decrement

int\* x = arr + 4;

int j = 5;

do{

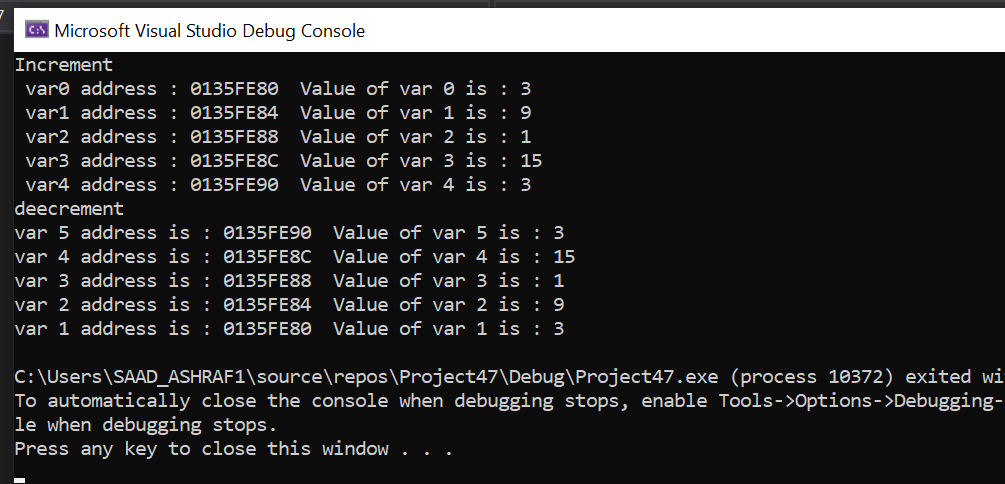
cout << "var " << j << " address is : "<< x <<" " << "Value of var " << j << " is : " << \*x << endl;

--x;

j--;

} while (j > 0);

}



Task 7:

#include<iostream>

using namespace std;

bool Prime\_Number(int\* p) {

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

if (\*p != i && i != 0 && i != 1) {

if (\*p % i == 0) {

cout << "The number is Not Prime number ";

return false;

}

}

}

cout << "The number is Prime number " << endl;

return true;

}

int main() {

cout << "If bool value is 1 then it is prime if it is 0 then it is not prime" << endl;

int o = 10;

cout << "Enter the NUmber ::";

cin >> o;

cout << "Your Bool Value is";

cout << endl;

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

cout << endl;

cout << Prime\_Number(&o);

}

