Task 1:

#include<iostream>

using namespace std;

int main()

{

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

int \*ptr;

ptr = a;

cout << "addresses and values with increment are : "<<endl;

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

{

cout << &ptr[i] <<" "<<ptr[i] << endl;

}

cout << " ------------OR---------------"<<endl;

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

{

cout << &\*(ptr + i) << " " << \*(ptr + i) << endl;

}

cout << "addresses and values with decrement are : " << endl;

for (int i = 4; i >= 0; i--)

{

cout << &ptr[i] << " " << ptr[i] << endl;

}

cout << #include<iostream>

using namespace std;

int main()

{

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

int \*ptr;

ptr = a;

cout << "addresses and values with increment are : "<<endl;

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

{

cout << &ptr[i] <<" "<<ptr[i] << endl;

}

cout << " ------------OR---------------"<<endl;

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

{

cout << &\*(ptr + i) << " " << \*(ptr + i) << endl;

}

cout << "addresses and values with decrement are : " << endl;

for (int i = 4; i >= 0; i--)

{

cout << &ptr[i] << " " << ptr[i] << endl;

}

cout << " ------------OR---------------"<<endl;

for (int i = 4; i >= 0; i--)

{

cout << &\*(ptr + i) << " " << \*(ptr + i)<< endl;

}

for (int i = 4; i >= 0; i--)

{

cout << &ptr[i] << " " << ptr[i] << endl;

}

cout << endl << endl;

//it checks whether the no. is equal to other no. or not

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

{

for (int j = i+1; j < 5; j++)

{

if (ptr[i] == ptr[j])

{

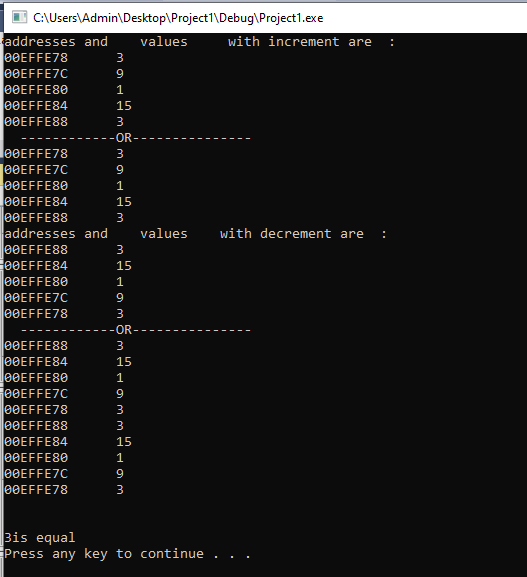
cout << (\*ptr + i)<< "is equal"<<endl;

}

}

}

system("pause");

}

Task 2:

#include<iostream>

using namespace std;

void primaryCheck(int num)

{

int j = 0;

bool prime = true;//using bool according to requirement

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

{

if (num%i == 0)//checks if no. is divisible to any no. or not

{

prime = false;

}

}

//2 and 1 are prime

if (j ==2 || j==1)

{

cout << "no. is prime";

}

if (prime)

{

cout << "it is prime ";

}

else

{

cout << "not a prime";

}

}

int main()

{

int num;

cin >> num;

//this func checks whether the no. is prime or not;

primaryCheck(num);

return 0;

}

------------------------------------------or----------------------------------------

#include<iostream>

using namespace std;

bool primaryCheck(int num)

{

int j = 0;

bool prime = true;

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

{

if (num%i == 0)

{

prime = false;

}

}

if (prime)

{

return true;

}

else

{

return false;

}

}

int main()

{

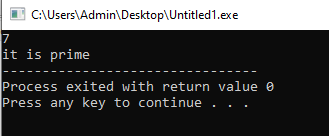
int num;

cin >> num;

primaryCheck(num);

return 0;

}



Task 3:

#include<iostream>

using namespace std;

int main()

{

int firstvalue = 15, secondvalue = 25;

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

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

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

p3 = &p1; // p3 = address of firstpointrer

p4 = &p2; // p4 = address of secondpointer

\*p1 = 100; // value pointed by p1 = 100

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

p1 = p2; // p1 = p2 (address of pointer is copied or not)

p3 = p4; // p3 =p4 (check address of pointer is copied or not)

\*p1 = 200; // value pointed by p1 = 200

#include<iostream>

using namespace std;

int main()

{

int firstvalue = 15, secondvalue = 25;

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

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

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

p3 = &p1; // p3 = address of firstpointrer

p4 = &p2; // p4 = address of secondpointer

\*p1 = 100; // value pointed by p1 = 100

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

p1 = p2; // p1 = p2 (address of pointer is copied or not)

p3 = p4; // p3 =p4 (check address of pointer is copied or not)

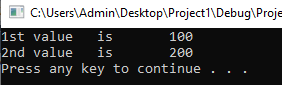
\*p1 = 200; // value pointed by p1 = 200

// print firstvalue, secondvalue

cout <<"1st value is "<< firstvalue<<endl;

cout << "2nd value is " << secondvalue << endl;

system("pause");

}

Task 4:

#include<iostream>

#include<iomanip>

using namespace std;

const int s = 10;

int main()

{

double a[s] = {0.0,1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9},var; //a

double \*nptr = &var; //b

//c

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

{

cout << setprecision(1) << right << fixed << a[i];

}

//d

nptr = &a[0];

nptr = a;

//e

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

{

cout << \*(nptr + i)<<endl;

}

//f

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

{

cout << \*(a + i);

}

//g

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

{

cout << setprecision(1) << right << fixed << nptr[i];

}

//h

var = a[4];

nptr = &a[4];

nptr = a+4;

//i

nptr = a;

cout <<"address is "<< &\*(nptr + 8);

cout << "value is " << \*(nptr + 8);

//j

nptr = &a[5];

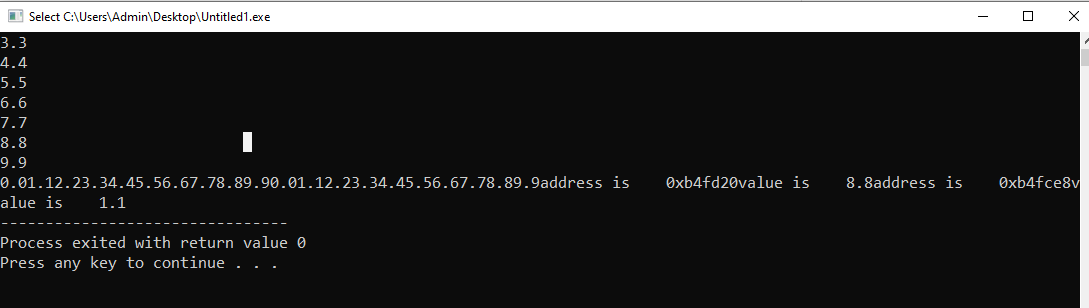
nptr -= 4;

cout << "address is " <<nptr ;

cout << "value is " << \*nptr;

system("pause");

}



Task 5:

#include<iostream>

using namespace std;

void print(int \*\*ptr2)

{

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

{

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

{

cout << \*(ptr2 + i) + j;

}

cout << endl;

}

}

int main()

{

int a[3][3] = { {1,2,3},{4,5,6},{7,8,9} };

int \*ptr = \*a;

int \*\*ptr2 = &ptr;

print(ptr2);

}

Task 6:

#include<iostream>

using namespace std;

void print(int a[5] )

{

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

{

cout << a[i] << endl;//printing array refering to pointer in which we passed the array values

}

}

int main()

{

int \*a,arr[5];

int\*ptr=new int[5];//assigns memory

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

{

cin >> ptr[i];//input in new array which the pointer point

}

print(ptr);//passing pointer in array to print it

system("pause");

}

