**Problem No: 01**

**Topic: Basic Structures: Sequences and Sum**

**Problem Title:**

Given a function f from {1, 2,...,n} to itself, determine whether i) f(x) = x2 ii) f(x) = x + 1 iii) f(x) = x3 + x2 + x + 1 is onto.

**Objectives:**

To learn onto function.

**Theory:**

A function is said to be onto if there exist an x for every y. When the range of the function is equal to codomain of the function then function is said to be onto or surjective function and if range is completely a subset of codomain then its is said to be into function.

**Source Code:**

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int func1 (int x){return x \* x;}

int func2 (int x){return x + 1;}

int func3 (int x){return (x \* x \* x) + (x \* x) + x + 1;}

void check\_func(int \*c, int \*r);

void print\_func(int \*d, int \*c, int \*f);

int n;

int main()

{

cout << "Enter n: ";

cin >> n;

int d[n], c[n], f1[n], f2[n], f3[n], i, j;

srand(unsigned(time(0)));

for(i = 0; i < n; i++){

j = (rand() % n) + 1;

d[i] = j;

c[i] = j;

}

for(i = 0; i < n; i++){

f1[i] = func1(d[i]);

f2[i] = func2(d[i]);

f3[i] = func3(d[i]);

}

print\_func(d, c, f1);

cout << "f(x) = x ^ 2 is";

check\_func(c, f1);

print\_func(d, c, f2);

cout << "f(x) = x + 1 is";

check\_func(c, f2);

print\_func(d, c, f3);

cout << "f(x) = x ^ 3 + x ^ 2 + x + 1 is";

check\_func(c, f3);

}

void check\_func(int \*c, int \*r)

{

int i, j, flag = 0;

for(i = 0; i < n; i++){

for(j = i; j < n; j++){

if(c[i] == r[j]){

flag = 0;

break;

}

else{

flag = 1;

}

}

if(flag == 1)

break;

}

if(flag == 0)

cout << " Onto Function\n";

else

cout << " Not Onto Function\n";

}

void print\_func(int \*d, int \*c, int \*f)

{

cout << "\nDomain " << "Co-Domain " << "Range\n";

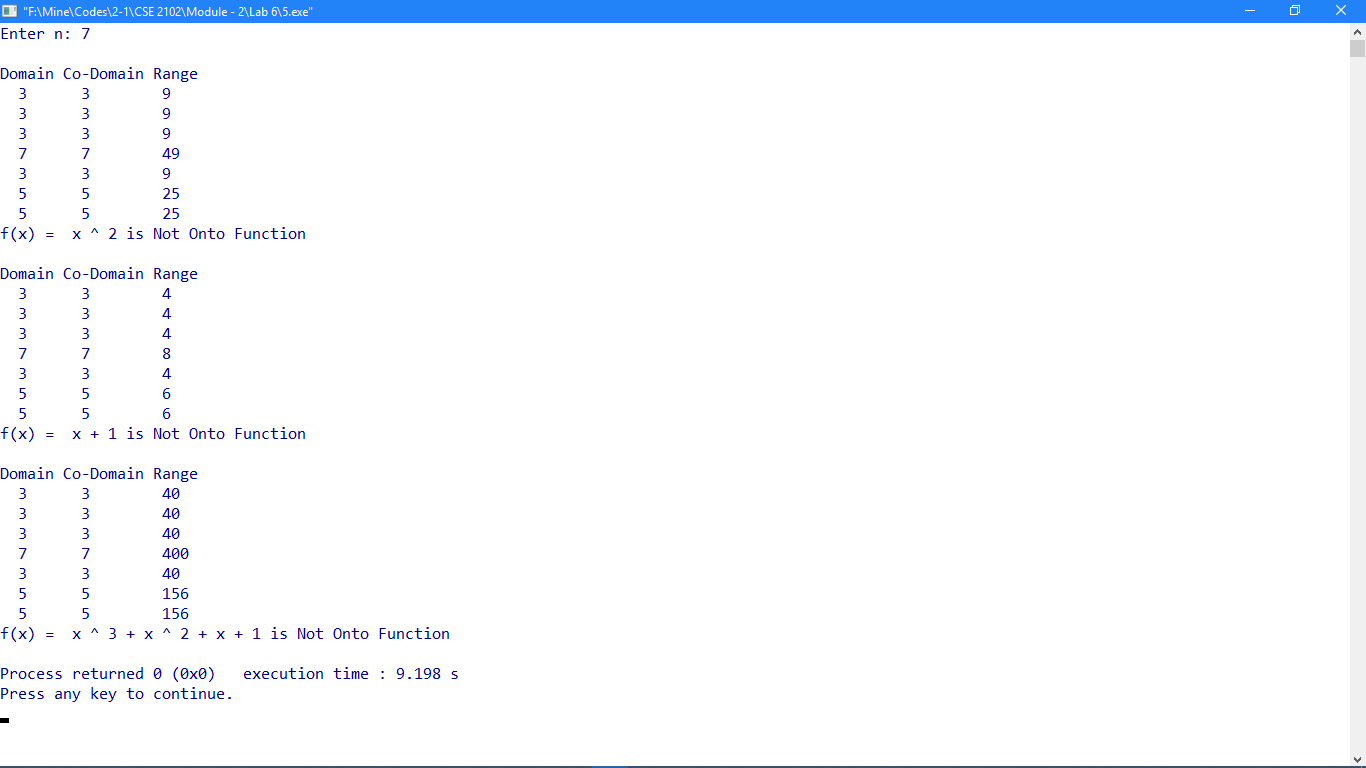
for(int i = 0; i < n; i++){

cout << " " << d[i] << "\t " << c[i] << "\t " << f[i] << endl;

}

}

**Output:**



**Problem No: 02**

**Topic: Basic Structures: Sequences and Sum**

**Problem Title:**

Check ⎣ 2x ⎦ = ⎣x⎦ + ⎣ ⎦ is true for integer number x = [-100 100] [Note: Check wide range of set if possible]

**Objectives:**

To learn about floor value.

**Source Code:**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

long int i, x1, x2;

cout << "Enter Range: \n";

cout << "from: ";

cin >> x1;

cout << "to: ";

cin >> x2;

cout << "\n|\_2x\_| = |\_x\_| |\_x+1/2\_| -this statement is ";

for (i = x1; i <= x2; i++) {

if ( floor(2 \* i) != floor(i)+floor( i + 1 / 2)) {

cout << "false.\n";

return 0;

}

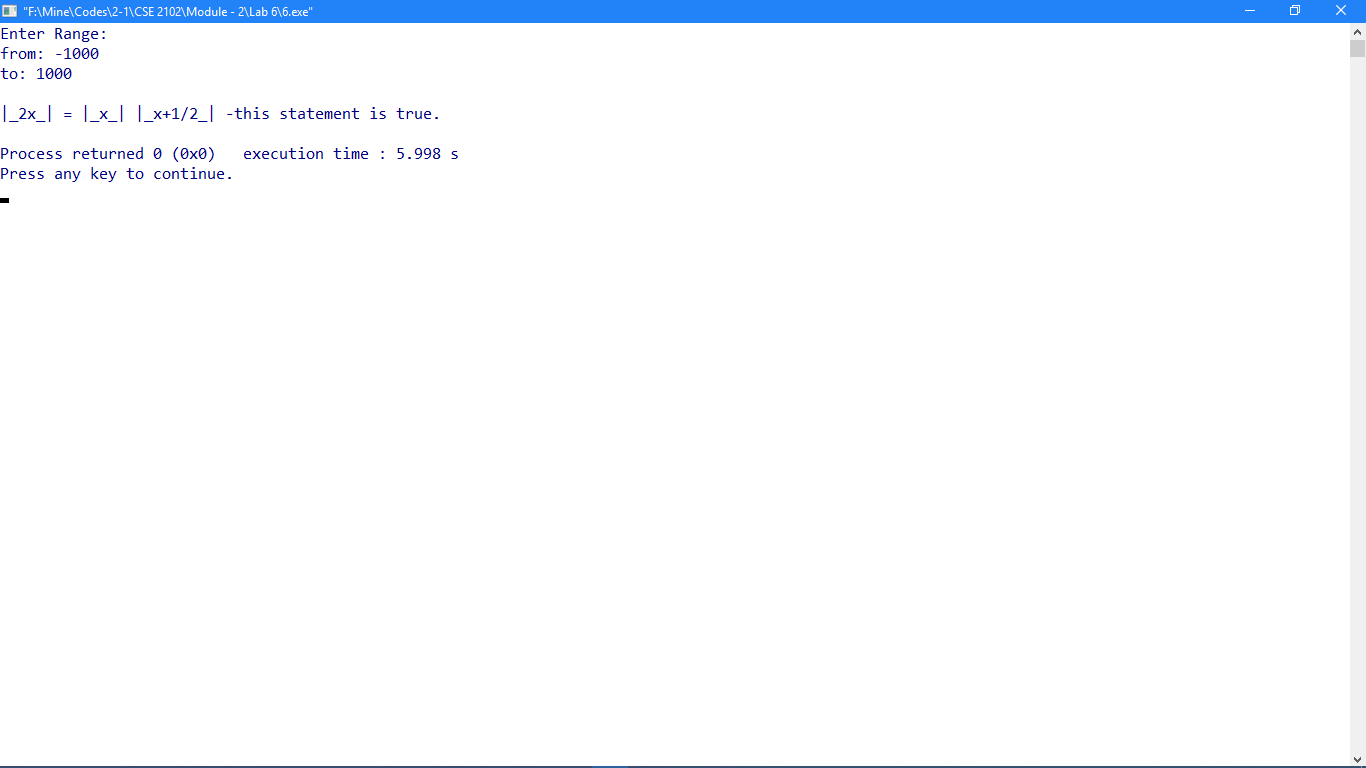
}

cout << "true.\n";

return 0;

}

**Output:**



**Problem No: 03**

**Topic: Basic Structures: Sequences and Sum**

**Problem Title:**

Find the following summation:

i) (a + nd ), where n = L to U

Where L<U, given L, U, a and d.

ii) arj where j = L to U

Where L<U, given L, U, a and r.

iii) (i + j) where i = L to U and j = L to U

Where L<U, given L, U.

**Objectives:**

To learn Summation.

**Source Code:**

#include <iostream>

#include <cmath>

using namespace std;

int L, U, a, d, r;

int func1();

int func2();

int func3();

int main()

{

cout << "Enter L: ";

cin >> L;

cout << "Enter U: ";

cin >> U;

cout << "Enter a: ";

cin >> a;

cout << "Enter d: ";

cin >> d;

cout << "Enter r: ";

cin >> r;

cout << "\nSum1: " << func1() << endl;

cout << "Sum2: " << func2() << endl;

cout << "Sum3: " << func3() << endl;

}

int func1()

{

int i, j, sum = 0;

for(i = L; i <= U; i++)

sum += a + (i \* d);

return sum;

}

int func2()

{

int i, j, sum = 0;

for(i = L; i <= U; i++)

sum += a \* pow(r, i);

return sum;

}

int func3()

{

int i, j, sum = 0;

for(i = L; i <= U; i++)

for(j = L; j <= U; j++)

sum += i + j;

return sum;

}

**Output:**

