using namespace std;

// Example program

#include <iostream>

#include <string>

float phasing(int m,int set\_size,uint bits, float twoClient\_cost\_Megbyte){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float res= ((m-1)\*twoClient\_cost\_Megbyte) + download\_cost\_MB;

return res;

}

// communication cost when client A and B engage the two-party protocol and

// then A and C, etc.

float Kamara(int m,int set\_size, uint bits){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float temp\_3 = (bits\* (m-1)\*3.5\* set\_size)/8000000;

float res = temp\_3 + download\_cost\_MB;

return res;

}

float basic\_Roi(int m, int set\_size, int bits, int BF\_bitSize){

cout<<"BF\_bitSize:"<<BF\_bitSize<<endl;

int temp\_1 = 2 \* BF\_bitSize/8000000;// this is for two clietn case

cout<<"temp\_1:"<<temp\_1<<endl;

int temp\_2= ((BF\_bitSize)\*(2\*bits))/8000000;

cout<<"temp\_2:"<<temp\_2<<endl;

int temp\_3 = temp\_1 + temp\_2;

cout<<"temp\_3:"<<temp\_3<<endl;

return temp\_3;

}

float Koles(int clients, int set\_size, int bits){//ccs'17

float m\_1 = 1184890.88; // 1.13 \* 1048576;

float B\_1 = 30;

float m\_2 = 178257.92; // 0.17 \* 1048576;

float B\_2 = 63;

float temp\_1 = bits \* ((m\_1 \* B\_1) + (m\_2 \* B\_2));

int m = m\_1 + m\_2;

float OPRF\_cost = m \* 500;

float res\_1 = clients\*(clients-1)\*(( temp\_1 + OPRF\_cost ));//step2

float res\_2 = (clients-1) \* (temp\_1 + OPRF\_cost);//step4

float res\_3 = res\_1 + res\_2;

float res\_4 = set\_size \* bits \* clients;// data downoad cost.

float res\_5 = res\_4 + res\_3; //total cost

return res\_5;

}

int main()

{

float res;

int number\_of\_clients = 2;

int set\_size = 1048576;

int bits = 128;

float twoClient\_cost\_Megbyte = 367.2; // 136.8 // 290.4 // 367.2

res = phasing(number\_of\_clients, set\_size, bits, twoClient\_cost\_Megbyte);

std::cout << "\nPhasing Result: "<<res<<endl;

float res\_2 = Kamara(number\_of\_clients, set\_size, bits);

std::cout << "\nKamara Result: "<< res\_2;

float Koles\_= Koles(number\_of\_clients, set\_size, bits);

std::cout << "\nKoles\_: "<< Koles\_/8000000<<endl;

}

pragma solidity ^0.5.0;

contract Cal{

uint public U\_PSI\_val;

uint public Kamara\_val;

uint public phasing\_;

uint public basic\_Roi\_;

uint d;

constructor () public{

d = 100;

}

function U\_PSI\_calc(uint m,uint h,uint bits)external returns (uint res){

U\_PSI\_val = bits\*(((m-1)\*(2+ h\*(2\*d+1))+1)+((m-1)\*( h\*(3 + 2\*d) + 1)) + h\*(2\*d + 1))+h\*5771;

return res;

}

function Kamara\_calc(uint m,uint set\_size,uint bits) external {

Kamara\_val = bits\*((2\*set\_size)+(m-1)\*set\_size+(set\_size/2)+(m\*set\_size));

}

function phasing(uint m,uint set\_size,uint bits, uint twoClient\_cost\_Megbyte) external {

uint twoClient\_cost\_bit = twoClient\_cost\_Megbyte \* 8000000;

phasing\_ = ((m-1)\*twoClient\_cost\_bit) + (m\*set\_size\*bits);

}

function basic\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

}

pragma solidity ^0.5.0;

contract Cal{

uint public U\_PSI\_val;

uint public Kamara\_val;

uint public phasing\_;

uint public basic\_Roi\_;

uint d;

constructor () public{

d = 100;

}

function U\_PSI\_calc(uint m,uint h,uint bits)external returns (uint res){

U\_PSI\_val = bits\*(((m-1)\*(2+ h\*(2\*d+1))+1)+((m-1)\*( h\*(3 + 2\*d) + 1)) + h\*(2\*d + 1))+h\*5771;

return res;

}

function Kamara\_calc(uint m,uint set\_size,uint bits) external {

Kamara\_val = bits\*((2\*set\_size)+(m-1)\*set\_size+(set\_size/2)+(m\*set\_size));

}

function phasing(uint m,uint set\_size,uint bits, uint twoClient\_cost\_Megbyte) external {

uint twoClient\_cost\_bit = twoClient\_cost\_Megbyte \* 8000000;

phasing\_ = ((m-1)\*twoClient\_cost\_bit) + (m\*set\_size\*bits);

}

function basic\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

function semi\_honest\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

//uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_1 = (m-1) \* (m-2) \* ((BF\_bitSize \* bits) + (2 \* BF\_bitSize \* bits));

//uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_2 = (m-1) \* ((BF\_bitSize \* bits) + (2 \* BF\_bitSize \* bits)); // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

}

using namespace std;

// Example program

#include <iostream>

#include <string>

float phasing(int m,int set\_size,uint bits, float twoClient\_cost\_Megbyte){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float res= ((m-1)\*twoClient\_cost\_Megbyte) + download\_cost\_MB;

return res;

}

// communication cost when client A and B engage the two-party protocol and

// then A and C, etc.

float Kamara(int m,int set\_size, uint bits){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float temp\_3 = (bits\* (m-1)\*3.5\* set\_size)/8000000;

float res = temp\_3 + download\_cost\_MB;

return res;

}

float basic\_Roi(int m, int set\_size, int bits, int BF\_bitSize){

cout<<"BF\_bitSize:"<<BF\_bitSize<<endl;

int temp\_1 = 2 \* BF\_bitSize/8000000;// this is for two clietn case

cout<<"temp\_1:"<<temp\_1<<endl;

int temp\_2= ((BF\_bitSize)\*(2\*bits))/8000000;

cout<<"temp\_2:"<<temp\_2<<endl;

int temp\_3 = temp\_1 + temp\_2;

cout<<"temp\_3:"<<temp\_3<<endl;

return temp\_3;

}

float Koles(int clients, int set\_size, int bits){//ccs'17

float m\_1 = 1.13 \* 1048576;

float B\_1 = 30;

float m\_2 = 0.17 \* 1048576;

float B\_2 = 63;

float temp\_1 = bits \* ((m\_1 \* B\_1) + (m\_2 \* B\_2));

int m = m\_1 + m\_2;

float OPRF\_cost = m \* 500;

float res\_1 = clients\*(clients-1)\*(( temp\_1 + OPRF\_cost ));//step2

float res\_2 = (clients-1) \* (temp\_1 + OPRF\_cost);//step4

float res\_3 = res\_1 + res\_2;

float res\_4 = set\_size \* bits \* clients;// data downoad cost.

float res\_5 = res\_4 + res\_3;// data downoad cost.

return res\_5;

}

int main()

{

float res;

int number\_of\_clients = 5;

int set\_size = 1048576;

int bits = 128;

float twoClient\_cost\_Megbyte = 367.2; // 136.8 // 290.4 // 367.2

res = phasing(number\_of\_clients, set\_size, bits, twoClient\_cost\_Megbyte);

std::cout << "\nPhasing Result: "<<res<<endl;

float res\_2 = Kamara(number\_of\_clients, set\_size, bits);

std::cout << "\nKamara Result: "<< res\_2;

float Koles\_= Koles(number\_of\_clients, set\_size, bits);

std::cout << "\nKoles\_: "<< Koles\_/8000000<<endl;

}

using namespace std;

// Example program

#include <iostream>

#include <string>

float phasing(int m,int set\_size,uint bits, float twoClient\_cost\_Megbyte){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float res= ((m-1)\*twoClient\_cost\_Megbyte) + download\_cost\_MB;

return res;

}

// communication cost when client A and B engage the two-party protocol and

// then A and C, etc.

float Kamara(int m,int set\_size, uint bits){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float temp\_3 = (bits\* (m-1)\*3.5\* set\_size)/8000000;

float res = temp\_3 + download\_cost\_MB;

return res;

}

float basic\_Roi(int m, int set\_size, int bits, int BF\_bitSize){

cout<<"BF\_bitSize:"<<BF\_bitSize<<endl;

int temp\_1 = 2 \* BF\_bitSize/8000000;// this is for two clietn case

cout<<"temp\_1:"<<temp\_1<<endl;

int temp\_2= ((BF\_bitSize)\*(2\*bits))/8000000;

cout<<"temp\_2:"<<temp\_2<<endl;

int temp\_3 = temp\_1 + temp\_2;

cout<<"temp\_3:"<<temp\_3<<endl;

return temp\_3;

}

float Koles(int clients, int set\_size, int bits){//ccs'17

float m\_1 = 1184890.88; // 1.13 \* 1048576;

float B\_1 = 30;

float m\_2 = 178257.92; // 0.17 \* 1048576;

float B\_2 = 63;

float temp\_1 = bits \* ((m\_1 \* B\_1) + (m\_2 \* B\_2));

int m = m\_1 + m\_2;

float OPRF\_cost = m \* 500;

float res\_1 = clients\*(clients-1)\*(( temp\_1 + OPRF\_cost ));//step2

float res\_2 = (clients-1) \* (temp\_1 + OPRF\_cost);//step4

float res\_3 = res\_1 + res\_2;

float res\_4 = set\_size \* bits \* clients;// data downoad cost.

float res\_5 = res\_4 + res\_3; //total cost

return res\_5;

}

int main()

{

float res;

int number\_of\_clients = 2;

int set\_size = 1048576;

int bits = 128;

float twoClient\_cost\_Megbyte = 367.2; // 136.8 // 290.4 // 367.2

res = phasing(number\_of\_clients, set\_size, bits, twoClient\_cost\_Megbyte);

std::cout << "\nPhasing Result: "<<res<<endl;

float res\_2 = Kamara(number\_of\_clients, set\_size, bits);

std::cout << "\nKamara Result: "<< res\_2;

float Koles\_= Koles(number\_of\_clients, set\_size, bits);

std::cout << "\nKoles\_: "<< Koles\_/8000000<<endl;

}

pragma solidity ^0.5.0;

contract Cal{

uint public U\_PSI\_val;

uint public Kamara\_val;

uint public phasing\_;

uint public basic\_Roi\_;

uint d;

constructor () public{

d = 100;

}

function U\_PSI\_calc(uint m,uint h,uint bits)external returns (uint res){

U\_PSI\_val = bits\*(((m-1)\*(2+ h\*(2\*d+1))+1)+((m-1)\*( h\*(3 + 2\*d) + 1)) + h\*(2\*d + 1))+h\*5771;

return res;

}

function Kamara\_calc(uint m,uint set\_size,uint bits) external {

Kamara\_val = bits\*((2\*set\_size)+(m-1)\*set\_size+(set\_size/2)+(m\*set\_size));

}

function phasing(uint m,uint set\_size,uint bits, uint twoClient\_cost\_Megbyte) external {

uint twoClient\_cost\_bit = twoClient\_cost\_Megbyte \* 8000000;

phasing\_ = ((m-1)\*twoClient\_cost\_bit) + (m\*set\_size\*bits);

}

function basic\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

}

pragma solidity ^0.5.0;

contract Cal{

uint public U\_PSI\_val;

uint public Kamara\_val;

uint public phasing\_;

uint public basic\_Roi\_;

uint d;

constructor () public{

d = 100;

}

function U\_PSI\_calc(uint m,uint h,uint bits)external returns (uint res){

U\_PSI\_val = bits\*(((m-1)\*(2+ h\*(2\*d+1))+1)+((m-1)\*( h\*(3 + 2\*d) + 1)) + h\*(2\*d + 1))+h\*5771;

return res;

}

function Kamara\_calc(uint m,uint set\_size,uint bits) external {

Kamara\_val = bits\*((2\*set\_size)+(m-1)\*set\_size+(set\_size/2)+(m\*set\_size));

}

function phasing(uint m,uint set\_size,uint bits, uint twoClient\_cost\_Megbyte) external {

uint twoClient\_cost\_bit = twoClient\_cost\_Megbyte \* 8000000;

phasing\_ = ((m-1)\*twoClient\_cost\_bit) + (m\*set\_size\*bits);

}

function basic\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

function semi\_honest\_Roi(uint m, uint set\_size, uint bits, uint BF\_bitSize) external{

//uint temp\_1 = m \*(m-1) \* BF\_bitSize \* bits; // total communication among clients-- note that the size of GBF is bits\*Bf\_bitsize

uint temp\_1 = (m-1) \* (m-2) \* ((BF\_bitSize \* bits) + (2 \* BF\_bitSize \* bits));

//uint temp\_2= m \* BF\_bitSize \*bits; // total communication between clients and the server

uint temp\_2 = (m-1) \* ((BF\_bitSize \* bits) + (2 \* BF\_bitSize \* bits)); // total communication between clients and the server

uint temp\_3 = m \* set\_size \* bits;//data download communication cost

basic\_Roi\_ = temp\_3 + temp\_1 + temp\_2;

}

}

using namespace std;

// Example program

#include <iostream>

#include <string>

float phasing(int m,int set\_size,uint bits, float twoClient\_cost\_Megbyte){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float res= ((m-1)\*twoClient\_cost\_Megbyte) + download\_cost\_MB;

return res;

}

// communication cost when client A and B engage the two-party protocol and

// then A and C, etc.

float Kamara(int m,int set\_size, uint bits){

float temp = (m\*set\_size\*bits);

float temp\_2 = 8000000;

float download\_cost\_MB = temp / temp\_2;

//cout<<"\n download\_cost\_MB: "<< download\_cost\_MB<<endl;

float temp\_3 = (bits\* (m-1)\*3.5\* set\_size)/8000000;

float res = temp\_3 + download\_cost\_MB;

return res;

}

float basic\_Roi(int m, int set\_size, int bits, int BF\_bitSize){

cout<<"BF\_bitSize:"<<BF\_bitSize<<endl;

int temp\_1 = 2 \* BF\_bitSize/8000000;// this is for two clietn case

cout<<"temp\_1:"<<temp\_1<<endl;

int temp\_2= ((BF\_bitSize)\*(2\*bits))/8000000;

cout<<"temp\_2:"<<temp\_2<<endl;

int temp\_3 = temp\_1 + temp\_2;

cout<<"temp\_3:"<<temp\_3<<endl;

return temp\_3;

}

float Koles(int clients, int set\_size, int bits){//ccs'17

float m\_1 = 1.13 \* 1048576;

float B\_1 = 30;

float m\_2 = 0.17 \* 1048576;

float B\_2 = 63;

float temp\_1 = bits \* ((m\_1 \* B\_1) + (m\_2 \* B\_2));

int m = m\_1 + m\_2;

float OPRF\_cost = m \* 500;

float res\_1 = clients\*(clients-1)\*(( temp\_1 + OPRF\_cost ));//step2

float res\_2 = (clients-1) \* (temp\_1 + OPRF\_cost);//step4

float res\_3 = res\_1 + res\_2;

float res\_4 = set\_size \* bits \* clients;// data downoad cost.

float res\_5 = res\_4 + res\_3;// data downoad cost.

return res\_5;

}

float Hazay\_basic(int clients, int set\_size, int pubKey\_bits){

float res\_inBits = (2 \* m - 1) \* set\_size \* pubKey\_bits;

float res\_inMB = res\_inBits / 8000000;

return res\_inMB;

}

int main()

{

float res;

int number\_of\_clients = 3;

int set\_size = 1048576;

int bits = 128;

int pubKey\_bits = 2048;

float twoClient\_cost\_Megbyte = 367.2; // 136.8 // 290.4 // 367.2

res = phasing(number\_of\_clients, set\_size, bits, twoClient\_cost\_Megbyte);

std::cout << "\nPhasing Result: "<<res<<endl;

float res\_2 = Kamara(number\_of\_clients, set\_size, bits);

std::cout << "\nKamara Result: "<< res\_2;

float Koles\_= Koles(number\_of\_clients, set\_size, bits);

std::cout << "\nKoles\_: "<< Koles\_/8000000<<endl;

float Hazay\_basic\_ = Hazay\_basic(clients, set\_size, pubKey\_bits);

std::cout << "\nHazay\_basic\_: "<< Hazay\_basic\_ <<endl;

}