#include <iostream>

#include <cstring>

using namespace std;

//REGISTERS:

int nAX, nBX, nCX;

int\* AX = &nAX;

int\* BX = &nBX;

int\* CX = &nCX;

//ZERO FLAG:

bool ZF = 0;

//MEMORY:

int nM1;

int nM2;

string nM3;

string nM4;

string nM5;

string nM6;

string nM7;

string nM8;

string nM9;

string nM10;

string nM11;

string nM12;

string nM13;

int\* M1 = &nM1;

int\* M2 = &nM2;

string\* M3 = &nM3;

string\* M4 = &nM4;

string\* M5 = &nM5;

string\* M6 = &nM6;

string\* M7 = &nM7;

string\* M8 = &nM8;

string\* M9 = &nM9;

string\* M10 = &nM10;

string\* M11 = &nM11;

string\* M12 = &nM12;

string\* M13 = &nM13;

//INSTRUCTION INDEX:

int instruction = 0;

bool running = true;

//OPERATIONS:

void SUB(int\* A, int B) {

\*A = \*A - B;

}

void MOV(int\* A, int B) {

\*A = B;

}

void DEC(int\* A) {

\*A = \*A - 1;

}

void JMP(int A) {

//ALWAYS JUMP

instruction = A - 1;

}

void JE(int A) {

if (ZF == 1) {

instruction = A - 1;

}

}

void JNE(int A) {

if (ZF == 0) {

instruction = A - 1;

}

}

void CMP(int\* A, int B) {

if (\*A == B) {

ZF = 1;

} else {

ZF = 0;

}

}

void PRT(string A) {

cout << A;

}

void END() {

running = false;

}

void DISPLAY() {

cout << "AX = " << \*AX << endl;

cout << "BX = " << \*BX << endl;

cout << "CX = " << \*CX << endl;

cout << "ZF = " << ZF << endl;

cout << "M1 = " << \*M1 << endl;

}

int main() {

//DEFINE BYTES

\*M1 = 1000;

\*M2 = 1234;

\*M3 = "\n";

\*M4 = "\*\* PLEASE TRY AGAIN\n";

\*M5 = "\*\* ENTER PIN: ";

\*M6 = "\*\* WITHDRAW? (1-YES, 0-NO): ";

\*M7 = "\*\* ENTER AMOUNT: ";

\*M8 = "\*\* SELECT OPTION (1-BALANCE, 2-WITHDRAW, 3-EXIT): ";

\*M9 = "\*\* ACCOUNT BALANCE: P";

\*M10 = "\*\* CASH WITHDRAWN: P";

\*M11 = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

\*M12 = "\*\* \*\*\*\*\*\* 8086 ATM \*\*\*\*\*\* \*\*\n";

\*M13 = "\*\* \*\*\n";

//PROGRAM

DISPLAY();

while (running) {

instruction++;

switch (instruction) {

case 1:

//START

MOV(CX, 3);

break;

case 2:

//LOAD CORRECT PIN

MOV(BX, \*M2);

break;

case 3:

//ENTER PIN

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M5);

break;

case 4:

//INPUT PIN

cin >> nAX;

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 5:

//VERIFY PIN

CMP(AX, \*BX);

break;

case 6:

//JUMP TO WITHDRAW?

JE(12);

break;

case 7:

//COUNT ATTEMPTS

DEC(CX);

break;

case 8:

//CHECK ATTEMPTS

CMP(CX, 0);

break;

case 9:

//JUMP TO EXIT

JE(46);

break;

case 10:

//PRINT PLEASE TRY AGAIN

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M4);

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 11:

//JUMP TO ENTER PIN

JNE(3);

break;

case 12:

//WITHDRAW?

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M6);

break;

case 13:

//INPUT WITHDRAW?

cin >> nAX;

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 14:

//CHECK WITHDRAW OPTION

CMP(AX, 1);

break;

case 15:

//JUMP TO WITHDRAW

JE(26);

break;

case 16:

//SELECT OPTION

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M8);

break;

case 17:

//INPUT OPTION

cin >> nAX;

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 18:

//CHECK FOR OPTION 1

CMP(AX, 1);

break;

case 19:

//JUMP TO BALANCE INQUIRY

JE(23);

break;

case 20:

//CHECK FOR OPTION 2

CMP(AX, 2);

break;

case 21:

//JUMP TO WITHDRAW

JE(26);

break;

case 22:

//JUMP TO EXIT

JMP(46);

break;

case 23:

//BALANCE INQUIRY

MOV(AX, \*M1);

break;

case 24:

//PRINT BALANCE

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M9);

PRT(to\_string(\*AX));

PRT(\*M3);

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 25:

//JUMP TO SELECT OPTION

JMP(16);

break;

case 26:

//WITHDRAW

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M7);

break;

case 27:

//INPUT AMOUNT

cin >> nAX;

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 28:

//LOAD BALANCE

MOV(BX, \*M1);

break;

case 29:

//LOOP CHECKER

MOV(CX, \*BX);

break;

case 30:

//SUBTRACT AMOUNT

SUB(BX, \*AX);

break;

case 31:

//INITIAL CHECKING

CMP(BX, 0);

break;

case 32:

//JUMP TO RECEIPT

JE(41);

break;

case 33:

//LOOP

DEC(BX);

break;

case 34:

//CHECK IF ZERO

CMP(BX, 0);

break;

case 35:

//JUMP TO RECEIPT

JE(41);

break;

case 36:

//COUNT LOOP

DEC(CX);

break;

case 37:

//CHECK LOOP COUNT

CMP(CX, 0);

break;

case 38:

//JUMP TO LOOP

JNE(33);

break;

case 39:

//PRINT PLEASE TRY AGAIN

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M4);

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 40:

//JUMP TO WITHDRAW

JE(26);

break;

case 41:

//RECEIPT

MOV(BX, \*M1);

break;

case 42:

//SUBTRACT AMOUNT AGAIN

SUB(BX, \*AX);

break;

case 43:

//UPDATE BALANCE

MOV(M1, \*BX);

break;

case 44:

//PRINT ACCOUNT BALANCE

PRT(\*M11);

PRT(\*M12);

PRT(\*M13);

PRT(\*M9);

PRT(to\_string(\*BX));

PRT(\*M3);

break;

case 45:

//PRINT CASH WITHDRAWN

PRT(\*M10);

PRT(to\_string(\*AX));

PRT(\*M3);

PRT(\*M13);

PRT(\*M11);

PRT(\*M3);

break;

case 46:

//EXIT

END();

break;

}

}

DISPLAY();

}