

#include <iostream>

#include <string>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

using namespace std;

class Player

{

public:

void newPlayer( int x, int y ) { setX(x); setY(y); }

void movePlayer() { enterStatus(); }

int getX() const { return posX\_\_; }

int getY() const { return posY\_\_; }

const char \*getItem() const { return item\_\_; }

private:

// settings

void enterStatus() { scanf("%d%d%s", &posX\_\_, &posY\_\_, itemList\_\_ ); fgetc(stdin);}

void setX( int x ) { posX\_\_ = x; }

void setY( int y ) { posY\_\_ = y; }

// variables

int posX\_\_;

int posY\_\_;

char itemList\_\_[100];

char \*item\_\_ = itemList\_\_;

};

class Customer

{

public:

// settings

void newCustomer( int id ) { id\_\_ = id; setOrder(); }

// get information

int getID() const { return id\_\_; }

char \*getOrder() const { return order\_\_; }

int getReward() const { return reward\_\_; }

private:

// settings

void setOrder() { scanf("%s%d", orderList\_\_, &reward\_\_); fgetc(stdin); } // in newCustomer

// variables

int id\_\_; // order id

int reward\_\_; // reward

char orderList\_\_[50]; // customer order string

char \*order\_\_ = orderList\_\_;

};

int main()

{

// basic variables

Customer customers[100];

Player me, partner;

char \*map = (char \*) new char[100];

int dishX, dishY, windowX, windowY, blueberryX, blueberryY, iceX, iceY;

char tableX[77], tableY[77], item[77];

/\*

\* first turn:

\* total number of customers, their orders and rewards

\*/

// total number of customers

int num\_all\_customers;

scanf("%d", &num\_all\_customers);

// orders and rewards

for (int i = 0; i < num\_all\_customers; i+=1)

{

customers[i].newCustomer(i);

}

// load kitchen

for (int i = 0; i < 7; i+=1)

{

for( int j = 0; j < 12; j+=1 )

{

char c = fgetc(stdin);

fprintf(stderr, "map:%c(%d,%d)\n", c, j, i );

if( c == '\n' )

{

;

}

else

{

map[i\*11+j] = c;

if( c == '0' )

{

me.newPlayer(j,i);

}

else if( c == '1' )

{

partner.newPlayer(j,i);

}

else if( c == 'D' )

{

dishX = j;

dishY = i;

}

else if( c == 'W' )

{

windowX = j;

windowY = i;

}

else if( c == 'B' )

{

blueberryX = j;

blueberryY = i;

}

else if( c == 'I' )

{

iceX = j;

iceY = i;

}

}

} // end j

} // end i

int dishblue = (dishX-blueberryX)\*(dishX-blueberryX) + (dishY-blueberryY)\*(dishY-blueberryY);

int dishice = (dishX-iceX)\*(dishX-iceX) + (dishY-iceY)\*(dishY-iceY);

/\*

\* In-game turns:

\* do the orders

\*/

while (1) {

// turns remaining

int turns;

scanf("%d", &turns );

// player status

me.movePlayer();

// partner status

partner.movePlayer();

// the number of tables in the kitchen that currently hold an item

int numItems;

scanf("%d", &numItems);

for (int i = 0; i < numItems; i+=1)

{

scanf("%d%d%s", &tableX[i], &tableY[i], &item[i] );

}

// ignore until wood 1 league

char oven\_contents[101];

int oven\_timer;

scanf("%s%d", oven\_contents, &oven\_timer);

// the number of customers currently waiting for food

int num\_customers;

scanf("%d", &num\_customers);

for (int i = 0; i < num\_customers; i+=1) {

customers[i].newCustomer(i);

}

/\*

\* make the moves

\*/

// MOVE x y

// USE x y

// WAIT

fprintf(stderr, "fprintf:%s\n", me.getItem() );

// holding a dish

if( me.getItem()[0] == 'D' )

{

if( strcmp(me.getItem(),"DISH-ICE\_CREAM-BLUEBERRIES") >= 0 )

{

printf("USE %d %d\n", windowX, windowY );

}

else if( strcmp(me.getItem(),"DISH-ICE\_CREAM") >= 0 )

{

if( (blueberryX>5) && (blueberryY>2) && (me.getY()<3) )

{

printf("MOVE 5 5\n" );

}

else

{

printf("USE %d %d\n", blueberryX, blueberryY );

}

}

else if( strcmp(me.getItem(),"DISH-BLUEBERRIES-ICE\_CREAM") >= 0 )

{

printf("USE %d %d\n", windowX, windowY );

}

else if( strcmp(me.getItem(),"DISH-BLUEBERRIES") >= 0 )

{

if( (iceX>5) && (iceY>2) && (me.getY()<3) )

{

printf("MOVE 5 5\n" );

}

else

{

printf("USE %d %d\n", iceX, iceY );

}

}

else

{

if( dishblue > dishice )

{

if( (iceY==0) || (iceY==6) || (iceX==0) || (iceX==10) )

{

printf("USE %d %d\n", iceX, iceY );

}

else

{

if( iceY == 2 )

{

printf("USE %d %d\n", iceX, iceY );

}

else if( iceX > 5 )

{

if( me.getY() < 4 )

{

printf("MOVE 9 4\n");

}

else

{

printf("USE %d %d\n", iceX, iceY );

}

}

else

{

if( me.getY() < 4 )

{

printf("MOVE 1 4\n");

}

else

{

printf("USE %d %d\n", iceX, iceY );

}

}

}

}

else

{

if( (blueberryY==0) || (blueberryY==6) || (blueberryX==0) || (blueberryX==10) )

{

printf("USE %d %d\n", blueberryX, blueberryY );

}

else

{

if( blueberryY == 2 )

{

printf("USE %d %d\n", blueberryX, blueberryY );

}

else if( blueberryX > 5 )

{

if( me.getY() < 4 )

{

printf("MOVE 9 4\n");

}

else

{

printf("USE %d %d\n", blueberryX, blueberryY );

}

}

else

{

if( me.getY() < 4 )

{

printf("MOVE 1 4\n");

}

else

{

printf("USE %d %d\n", blueberryX, blueberryY );

}

}

} // end blueberry center

} // end dishblue <= dishice

}

} // end "holding a dish"

// get the dish first

else

{

printf("USE %d %d\n", dishX, dishY );

}

}

delete [] map;

return 0;

}