One and a half digger

(x+y-1)/y

The king of the castle Einavzan received a report that the troops of a formidable enemy were coming to his monastery. The inhabitants of the kingdom can only have one advantage - to lock themselves in a castle and repel enemy attacks from the fortress walls until the help of allies comes up. To increase the chances of success, the king ordered a moat to be dug around the castle, but the time left is catastrophically short, in two days the enemy will be too close. Help the king understand how many people need to be sent to prepare the ditch, while the rest of the residents of the surrounding villages will move to the castle and prepare supplies, if in a day one excavator can dig a ditch 3 meters long, and the length of the supposed ditch is 1400 meters.

Output format

ceil

One number is the required number of diggers.

36

37

```
int diggerNum;
int ditchLength;
int daysNum = 2;
int ditchSpeed = 3;

std::cin >> ditchLength;

// 1400 / 3 = ~ 466 / 2 = 233;
// 36 / 3 / 2 = 6;
  if (ditchLength % (ditchSpeed * daysNum) == 0 ) {
     diggerNum = ditchLength / ditchSpeed / daysNum;
  }
  else
  {
     diggerNum = ditchLength / ditchSpeed / daysNum + 1;
  }

std::cout << "Takes " << diggerNum << " diggers." << std::endl;
```

```
int main() {
```

cout<<ceil(ditchLength/(ditchSpeed * daysNum))<<endl;

```
int diggerNum;
int ditchLength;
int daysNum = 2;
int ditchSpeed = 3;

std::cin >> ditchLength;

// 1400 / 3 = ~ 466 / 2 = 233;
// 36 / 3 / 2 = 6;
     diggerNum = (ditchLength + ditchSpeed * daysNum - 1) / (ditchSpeed * daysNum);
     // ( x + y - 1) / y;
     // adds "buffer" of whatever y is ( -1 ) divided by y , if it needs buffer then y / y = 1
     // otherwise, if it's y - 1 / y then the "additional" 1 is divided and disappears (no remainder)

std::cout << "Takes " << diggerNum << " diggers." << std::endl;
```