Random number generation

Algorithms and programming (Paper 2) · revision notes

Programs can generate random numbers, for example to simulate a dice or pick a random question. In ERL, random(a, b) gives a random number between a and b inclusive. Expect questions on the possible values and short programs such as games and simulations.

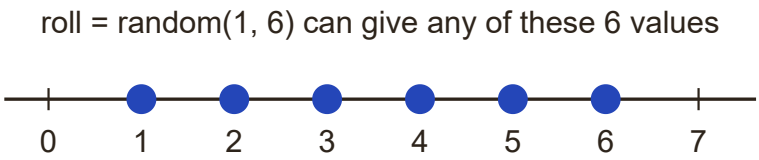
Key facts

- random(a, b): random number from a to b, both included
- random(1.0, 5.0): a random real number
- Store the number in a variable if you need the same value again
- random(1, 6): a dice roll (1, 2, 3, 4, 5 or 6)
- Random array item: array[random(0, n − 1)] for n items
- Uses: games, simulations, random choices, test data

Notes

Generating random numbers in ERL

- `roll = random(1, 6)` stores a random whole number from 1 to 6. Both limits are included.



random(1, 6): both limits are included

- `x = random(-1.0, 10.0)` gives a random real number between −1.0 and 10.0.
- Each call to random() gives a new number, so generate the number once and store it in a variable if you need to use the same value more than once.
- To pick a random item from an array with 8 elements (indexes 0 to 7): `tip = tips[random(0, 7)]`.

Uses of random numbers

- Games: dice rolls, shuffling cards, where an enemy appears.
- Simulations: modelling events that happen by chance, such as the weather or queues.
- Choosing a random item, such as a quiz question, a password character or a prize winner.
- Generating large amounts of test data.

Example: a dice simulation

- Count how many sixes appear in 100 rolls:

```
sixes = 0
for i = 1 to 100
   roll = random(1, 6)
   if roll == 6 then
      sixes = sixes + 1
   endif
next i
print(sixes)
```

- The random number is generated **inside** the loop, so each roll is a new random number. The count will vary each time the program runs.

How to answer each type of question

State the possible values 1 to 2 marks **Grade 4**

1. The smallest and largest values are the two limits in random(a, b).
2. Apply any calculation to both limits.
3. Count the whole numbers from a to b inclusive: b − a + 1.

Example. `score = random(3, 8) * 2`
(a) State the smallest and largest values score could have. [2]
(b) State how many different values random(3, 8) can give. [1]

Model answer.

(a) Smallest 6 (1); largest 16 (1)
(b) 6 (1)

Pick a random item from an array 2 marks **Grade 7**

1. Work out the index range: 0 to (number of items − 1).
2. Use random() as the index and output or store the element.

Example. The array `tips` holds 8 strings. Write one line of code that outputs a tip chosen at random. [2]

Model answer.

```
print(tips[random(0, 7)])
```

random(0, 7) to give a valid index (1); used as the index of tips and output (1).

Write a game or simulation

5 to 6 marks | **Grade 8**

1. Generate the random number once, before the loop, if it must stay the same (such as a secret number).
2. Use a condition-controlled loop for repeated guesses.
3. Use selection to give feedback and a counter to keep track.
4. Output the final result after the loop.

Example. Write a program for a guessing game. The computer chooses a random whole number from 1 to 50. The user keeps guessing until they are correct. After each wrong guess, output `Higher` or `Lower`. At the end, output the number of guesses taken.
You must use either OCR Exam Reference Language or a high-level programming language that you have studied. [6]

Model answer.

```
secret = random(1, 50)
guesses = 0
do
   guess = int(input("Enter your guess"))
   guesses = guesses + 1
   if guess < secret then
      print("Higher")
   elseif guess > secret then
      print("Lower")
   endif
until guess == secret
print("Correct in " + str(guesses) + " guesses")
```

Random number from 1 to 50 generated once, before the loop (1); loop repeats until the guess equals the number (1); guess input and cast to an integer inside the loop (1); outputs Higher or Lower correctly (1); counts the guesses (1); outputs the count after the loop (1).

Don’t lose marks

- Generating the secret number inside the loop, so it changes after every guess.
- Using random(1, 8) as an index for an 8-item array: index 8 does not exist and index 0 is never picked.
- Calling random() twice when the same value was needed in two places.
- Forgetting that the limits are included, for example saying random(1, 6) can only give 1 to 5.

More tips

Memory tricks

- random(a, b) includes both ends, like a dice numbered a to b.
- Array index range = 0 to length − 1, so a random index is random(0, length − 1).
- Secret number: pick it once, outside the loop. New dice roll: inside the loop.

Exam technique

- Check the range in the question: 'from 1 to 50' means random(1, 50).
- Explain in words if needed: 'generates a random whole number from 1 to 50 and stores it in secret'.
- Random output cannot be predicted, so a trace question will usually tell you which random values to use.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 4 — **State the range of values random() can give**
random(1, 6) gives a whole number from 1 to 6 inclusive.

Grade 5 — **Give uses of random numbers**
Games, simulations, choosing random items and generating test data.

Grade 6 — **Write code that uses random numbers**
Store a random number in a variable and use it in selection or a loop.

Grade 7 — **Pick a random element from an array**
For an array of n items use the index random(0, n − 1).

Grade 8 — **Write a simulation or game using random**
Combine random numbers with loops, counters and selection correctly.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.