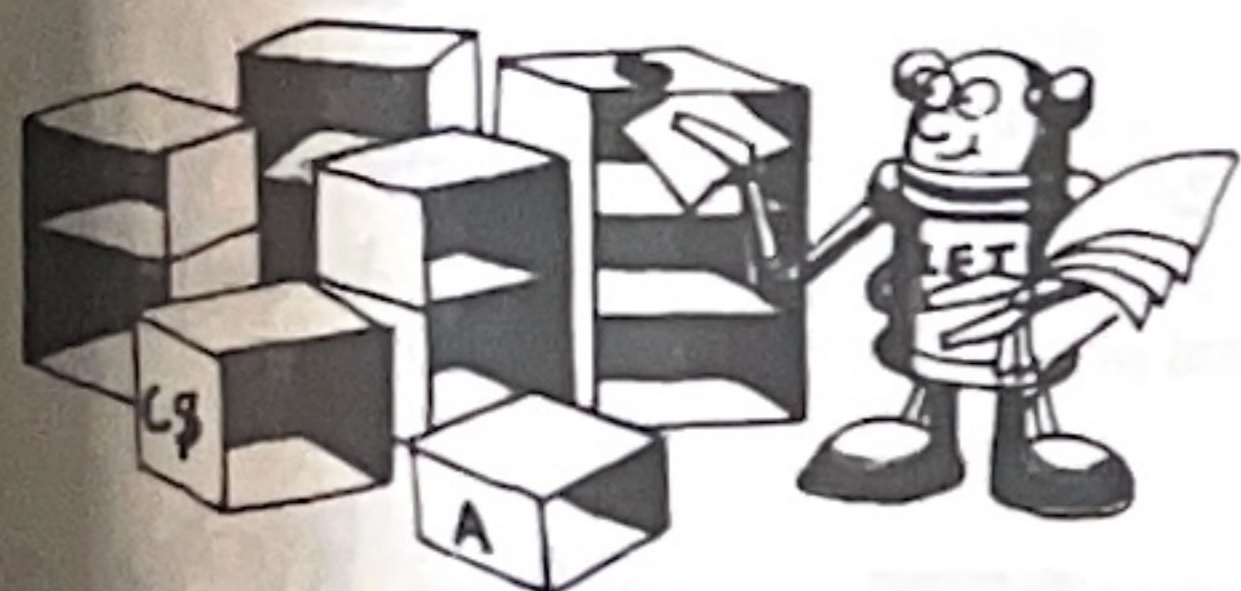


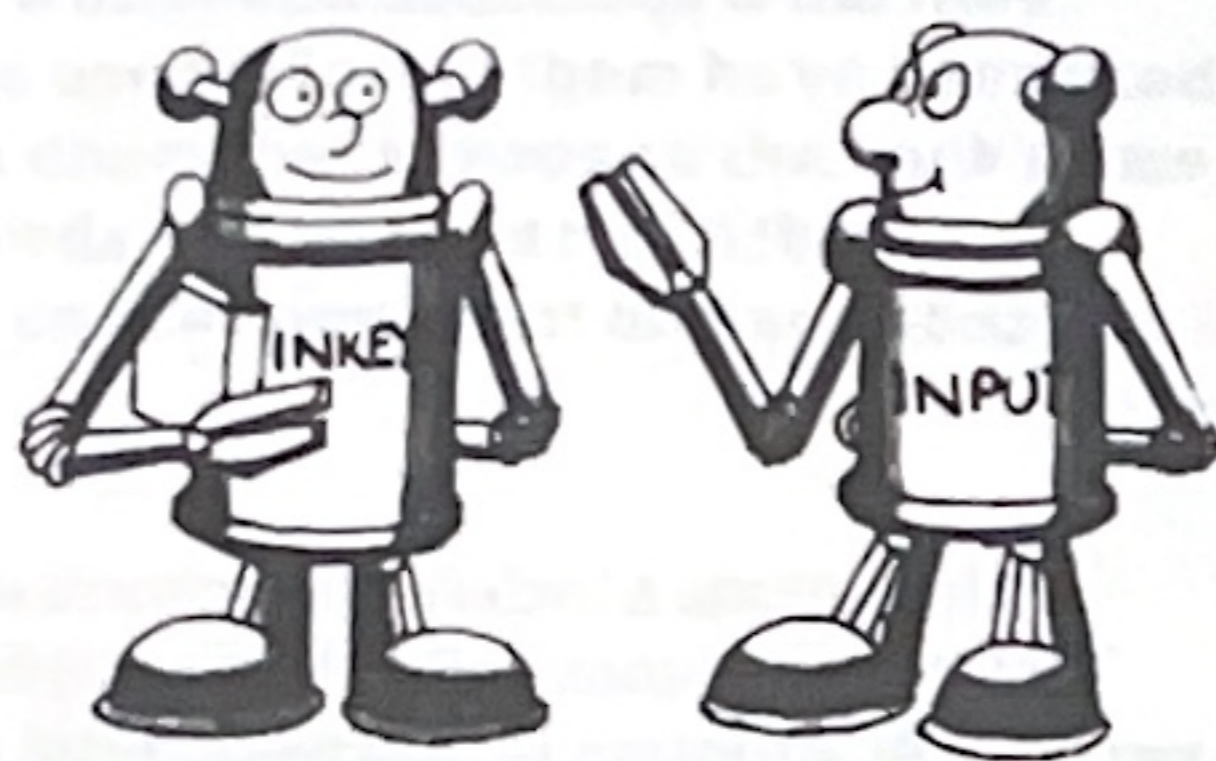
# Writing your own programs

As you work through the games in the book, you will probably find yourself making more and more changes to them and eventually wanting to write new games of your own. On these two pages you will find some hints on how to set about doing this.

Before you start, it is a good idea to stop and think about what your computer can and cannot do.



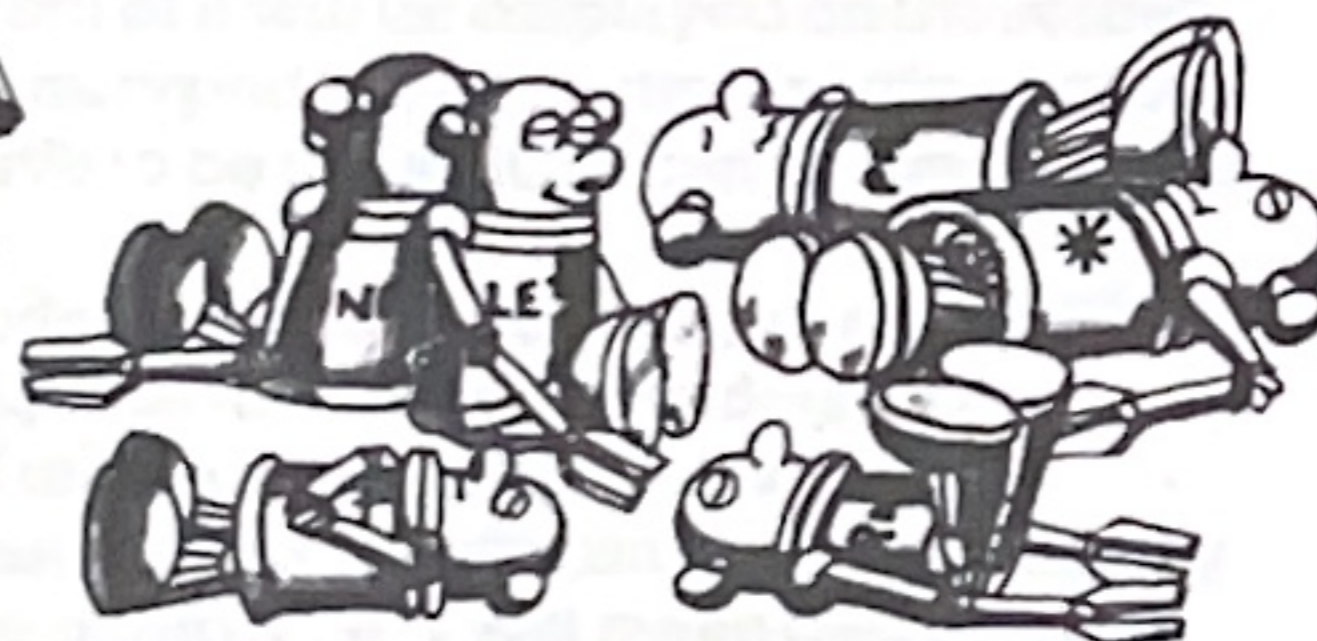
\*It can store information



\*It can ask you for information.



\*It can select numbers at random by using RND.



\*It cannot do anything unless you tell it to.



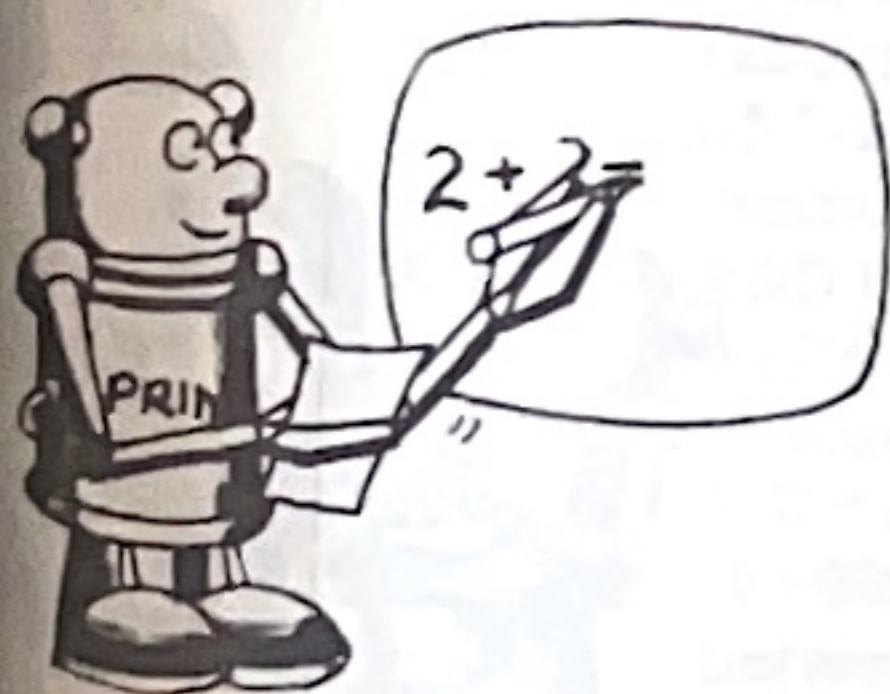
\*Provided you use its language correctly, it can do only *exactly* what you tell it, even if it is silly.

Remember, when you are trying to work out a game, not to include anything which your computer won't be able to do.

\*It can do calculations.



\*It can make decisions by comparing items of information in various ways.



\*It can tell you the results of its calculations and decisions and also what is stored in its memory.

## Planning a game

Before you can tell the computer how to play your game, you must know exactly how to play it and what the rules are yourself. The computer will need a series of simple logical instructions, so work out your game in your head or on paper first and then break it down into simple steps.

Next write a plan (in English - don't try to use BASIC yet) of all the stages of the game in order.

Here is a plan for a simple shooting game, such as firing cannon balls at a pirate ship or shooting laser beams at an alien invader, to give you an idea.

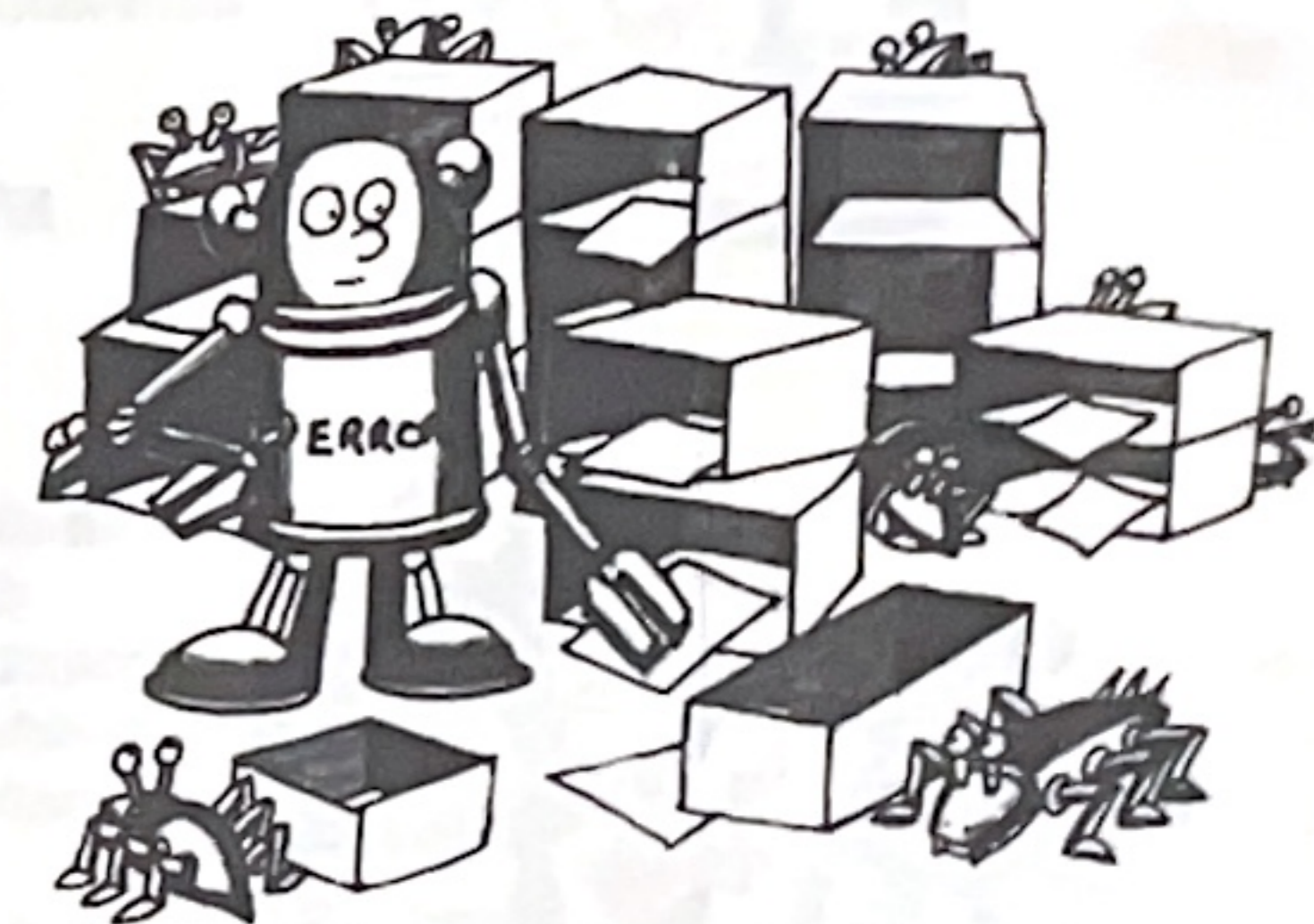
### PLAN

- 1) PRINT TITLE AND INSTRUCTIONS
- 2) CHOOSE A TARGET FOR THIS GAME
- 3) BEGIN A LOOP TO GIVE THE PLAYER N GOES
- 4) GET A SHOT FROM THE PLAYER
- 5) CHECK IF SHOT WAS ON TARGET
- 6) PRINT MESSAGE DEPENDING ON ACCURACY OF SHOT
- 7) GO BACK FOR ANOTHER GO IF SHOT WAS UNSUCCESSFUL

## Writing the program

The next stage is to convert your plan into BASIC. Each step in your plan may need several lines in BASIC. Don't forget to leave gaps when numbering your program lines so you can go back and add extra ones if you need to.

Do a first draft of the program on paper first and then start testing on the computer. Your computer will spot errors much more quickly than you will see them yourself and may give you a clue as to what is wrong. Remember that debugging programs is a long, tedious process even for expert programmers, so don't expect to get yours right first time.



Once you have got the core of the program working, you can add to it. Scoring, extra comments, more targets etc. can all be incorporated later. You could add sections from the programs in this book to your games.

Don't expect to be able to write exciting and original games straight away. Keep your ideas very simple and be prepared to adapt them as you go along. You may find you have included something in your game which is easy for humans to do but very difficult for a computer. As you get more experienced you will begin to know instinctively what your computer can do and find it easier to write programs for it.