Design to code

Design the algorithm then translate it into a language

Description

Pack my school bag. If rain is forecast, pack a raincoat. Otherwise, pack a hat. Print the bag contents.

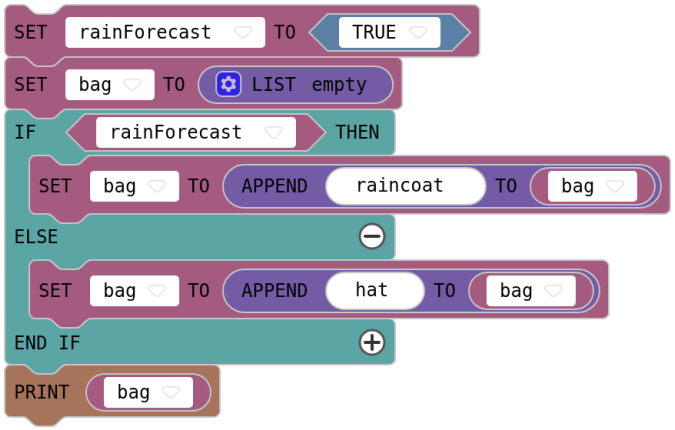
Use a true or false forecast and start with an empty bag.

What else do you have to pack today? What decisions are involved?

Design

Pseudly

Blocks



Generate

Generated pseudocode

```
SET rainForecast TO TRUE
SET bag TO []
IF rainForecast THEN
    SET bag TO (APPEND "raincoat" TO bag)
ELSE
    SET bag TO (APPEND "hat" TO bag)
END IF
PRINT bag
```

Translate

Implement in a language

JavaScript

```
const rainForecast = true;
let bag = [];
if (rainForecast) {
  bag = [...bag, "raincoat"];
} else {
  bag = [...bag, "hat"];
}
console.log(bag);
```

Python

```
rainForecast = True
bag = []
if rainForecast:
    bag.append("raincoat")
else:
    bag.append("hat")
print(bag)
```

C++

```
bool rainForecast = true;
std::vector<std::string> bag;
if (rainForecast) {
  bag.push_back("raincoat");
} else {
  bag.push_back("hat");
}
for (const auto& item : bag)
  std::cout << item << '\n';
```

Test and refine

Simplified rule: no rain means choose a hat. Python is our main example; the algorithm also has other implementations.

C++ excerpt: inside main, with iostream, string and vector included. Full programs are on the companion page.