

Học xong bài này, em sẽ:

Sử dụng được một số hàm thư viện có sẵn của Python để giải bài toán thực tế.

Nhiệm vụ và yêu cầu kết quả

Thực hiện phương án sử dụng tối đa các hàm đã có sẵn trong Python để tạo ra SP#2: Thay vì ưu tiên viết trực tiếp các câu lệnh chi tiết như trong Bài 11 và Bài 12, bài này yêu cầu thay thế bằng việc gọi sử dụng các hàm có sẵn trong Python mỗi khi có thể.

Nhiệm vụ 1. Viết phiên bản thứ hai cho hàm thực hiện phân tích dãy điểm

Yêu cầu:

Viết chương trình thực hiện hàm **ptDiem** theo phương án sử dụng tối đa các hàm có sẵn trong Python và chạy thử kiểm tra.

Hướng dẫn thực hiện:

- Sử dụng hàm **sum** để tính tổng và điểm trung bình.
- Gọi hàm Python thực hiện sắp xếp thứ tự tăng dần (không giảm); sau khi sắp xếp thì tìm được ngay **max**, **min**.
- Dãy số đã sắp thứ tự tăng dần (không giảm) nên có thể dùng hàm **bisect_left** (trong mô đun **bisect**) tìm được các vị trí phân chia dãy điểm thành 4 đoạn điểm: Chưa đạt, Đạt, Khả và Tốt. Từ đó tính được số lượng điểm theo từng mức xếp hạng.



```
File Edit Format Run Options Window Help
1 def ptDiem_bisect(ds):
2     ...
3     chuadat = bisect_left(ds,DAT) #Số điểm ở mức chưa đạt
4     dat = bisect_left(ds,KHA) - chuadat #Số điểm ở mức đạt
5     ...
```

Hình 1. Trích một số câu lệnh định nghĩa hàm **ptDiem_bisect**

Nhiệm vụ 2. Viết phiên bản thứ hai của hàm **ptMonHoc**

Yêu cầu:

Trong hàm **ptMonHoc** thay vì sử dụng **quickSort_tuple_down** cải biên từ hàm **quicksort** hãy gọi sử dụng các hàm có sẵn trong Python.

Hướng dẫn thực hiện:

– Python có sẵn phương thức **sort**, hàm **sorted** với tham biến **key = lambda** để sắp xếp danh sách các cặp hay các bộ nhiều thành phần, theo giá trị của một thành phần bất kì trong bộ. Cách gọi sử dụng để sắp xếp danh sách các cặp (*tên, điểm*) theo thứ tự điểm giảm dần theo mẫu tham khảo Hình 2:

```
File Edit Format Run Options Window Help
1 dsGiam = sorted(ds, reverse = True, key = lambda banghi: banghi[1])
```

Hình 2. Minh họa thực hiện sắp xếp bằng hàm **sorted**

- Trong thân hàm **ptMonHoc** sửa lại lời gọi hàm thực hiện việc sắp xếp, sử dụng hàm của Python;
- Để dùng hàm **bisect_left** cần sắp thứ tự tăng dần;
- Khi dãy đã sắp thứ tự tăng dần thì chỉ cần viết lại theo thứ tự đảo ngược để có dãy giảm dần.



Chạy chương trình thực hiện hàm **ptDiem_bisect** vừa hoàn thành với đầu vào là kết quả môn Tin học của lớp em và cho biết: Số lượng điểm của mức nào là nhiều nhất? Nên xếp hạng kết quả học tập môn Tin học chung của cả lớp ở mức nào?

BÀI TÌM HIỂU THÊM

ZEN OF PYTHON

Kĩ sư phần mềm Tim Peters đã tập hợp 19 “nguyên tắc hướng dẫn” để lập trình “tốt” và đăng lên danh sách gửi thư của nhóm phát triển Python vào năm 1999. Những nguyên tắc này đã ảnh hưởng đến thiết kế của ngôn ngữ lập trình Python. Danh sách của Peters đề nghị nguyên tắc thứ 20 “để Guido điền vào”, dành cho Guido van Rossum – tác giả ban đầu của ngôn ngữ Python.

“Zen of Python” của Peters đã được đưa vào mục số 20 trong “Bộ Đề nghị Cải tiến Python chính thức” và được phát hành công cộng. Trong cửa sổ trình thông dịch Python, có thể gõ nhập “import this” để hiển thị toàn văn. Dưới đây phỏng dịch một số điểm chính:

- Trường minh tốt hơn hiểu ngầm.
- Đơn giản tốt hơn phức hợp.
- Phức hợp tốt hơn phức tạp.
- Tuân tự tốt hơn lỏng lẻo.
- Viết câu lệnh rõ ràng tốt hơn viết rút ngắn cô đặc.
- Chương trình dễ đọc là quan trọng.
- Không “im lặng bỏ qua” tiềm năng lỗi.