

انسان یا دانشجو، مسئله این است

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک کلاس با نام `Person` ایجاد کنید که شامل ویژگی‌ها و متدهای زیر باشد:

ویژگی‌ها:

- `name (str)`: نام شخص
- `age (int)`: سن شخص
- `gender (str)`: جنسیت شخص ("Male" یا "Female")

متدها:

- `__init__(self, name, age, gender)`: ویژگی‌های `name`، `age` و `gender` را مقداردهی

اولیه می‌کند

- `get_name(self)`: ویژگی `name` را برمی‌گرداند

- `get_age(self)`: ویژگی `age` را برمی‌گرداند

- `get_gender(self)`: ویژگی `gender` را برمی‌گرداند

- `__str__(self)`: یک رشته بازگشتی از شخص با فرمت `"Name: [name], Age: [age], Gender: [gender]"`

یک زیرکلاس با نام `Student` ایجاد کنید که از کلاس `Person` ارث‌بری کرده و شامل ویژگی‌ها و متدهای زیر باشد:

ویژگی‌ها:

- `student_id (str)`: یک شناسه یکتا برای دانش‌آموز
- `courses (list)`: یک لیست از نام دروسی که دانش‌آموز در آن‌ها ثبت‌نام کرده است

متدها:

- `__init__(self, name, age, gender, student_id)`: ویژگی‌های `name`، `age`، `gender` و `student_id` و یک لیست خالی `courses` را مقداردهی اولیه می‌کند
 - `add_course(self, course)`: یک نام درس را به لیست `courses` اضافه می‌کند
 - `remove_course(self, course)`: یک نام درس را از لیست `courses` حذف می‌کند
 - `get_courses(self)`: لیست `courses` را برمی‌گرداند
 - `__str__(self)`: یک رشته بازگشتی از دانش‌آموز با فرمت `"Name: [name], Age: [age], Gender: [gender], Student ID: [student_id], Courses: [courses]"` را برمی‌گرداند
- نشان می‌دهد.

ورودی

در خط اول به شما چهار ویژگی `name`, `age`, `gender`, `student_id` که با اسپیس از هم جدا شده‌اند داده می‌شود.

در خط بعدی عدد `n` که تعداد دروس دانشجو است داده می‌شود.

در `n` خط بعدی در هر خط نام یک درس که یکی از دروس دانشجو است داده می‌شود.

خروجی

در خروجی برنامه کافیت شی `student` ساخته شده را `print` کنید.

ورودی نمونه ۱

```
Ali 21 Male 400440000
3
Calculus 2
Advanced Programming
Linear Algebra
```

خروجی نمونه ۱

Name: Ali, Age: 21, Gender: Male, Student ID: 400440000, Courses: ['Calculus 2', 'Adva