

پروژه های کلاس پایتون

جلسه 1

1. برنامه ای بنویسید که تعداد روز را بگیرد و تعداد سال، هفته و روز را برگرداند. (راهنمایی مثلا 375 را بگیرد و خروجی بدهد که 375 معادل 1 سال و 1 هفته و 3 روز است)(هر سال را 365 روز حساب کنید)
2. کدی بنویسید که دمای ورودی را به فارنهایت گرفته و تبدیل به سلسیوس کند

جلسه 2

1. پیاده سازی تصویر درخت تصمیم آپلود شده
2. BMI خود را محاسبه کرده و میزان اضافه وزن یا کمبود وزن خود را تعیین کنید
3. برنامه ای بنویسید که یک عدد صحیح از کاربر بگیرد و بررسی کند که آیا فرد است یا زوج.
4. در تمرین 3 اگر عدد مورد نظر زوج باشد بررسی کند که اگر بر 5 بخش پذیر باشد یک عدد دیگر از کاربر بگیرد و عدد جدید را با عدد قبلی جمع کند.

جلسه 3

1. برنامه نویسی ترسیم تصاویر موجود در سورس کد
2. فرض کنید الان 80 کیلو گرم هستید اگر شما روی کره ماه خودتان را وزن کنید وزنتان 16.5 درصد وزن کره زمین خواهد شد یعنی برای محاسبه وزن کره ماه وزن کره زمین در 0.165 ضرب می شود. اگر شما تا 15 سال آینده هر سال 1 کیلو گرم به وزنتان اضافه شود و هر سال خوتان را در کره ماه وزن کشی کنید ترازوی کره ماه هر سال چه عددی را نمایش خواهد داد؟
3. یک عدد صحیح از کاربر بگیرید و میانگین اعداد از صفر تا آن عدد را محاسبه کنید (از حلقه استفاده کنید)
4. یک عدد صحیح از کاربر بگیرید و فاکتوریل آن عدد را محاسبه کنید (از طریق حلقه).
5. برنامه ای بنویسید یک عدد صحیح از کاربر بگیرد و مجموع اعداد فرد از صفر تا آن عدد را محاسبه کرده و در مجموع اعداد مضرب 5 از صفر تا آن عدد ضرب کند.

جلسه 4

1. اسم خودتان را در یک متغیر بنویسید و برعکس کنید (از دو روش استفاده کنید 1- از روش اسلایسینگ 2- از طریق حلقه ها)
2. برنامه ای بنویسید مقداری مثل 24hours2days بگیرد و مقدار ارقام و رشته ها را در متغیرهای جداگانه ای ذخیره کند.
3. برنامه ای بنویسید تا 10 ورودی عدد صحیح از کاربر گرفته و آنها را در هم ضرب کند.
4. برنامه بنویسید که پترنی مانند زیر تولید کند (حرف E انگلیسی)

*

*

*

*

5. از کاربر بخواهید که 10 عدد را وارد کنید و بدون استفاده از توابع min و max مینیمم و ماکزیمم آنها را حساب کنید

6. برنامه‌ای بنویسید که اعداد ۱ تا ۲۰ را چاپ کند، اما فقط اعداد زوج را نشان دهد. از دستور continue استفاده کنید تا از چاپ اعداد فرد صرف‌نظر شود.

7. برنامه‌ای بنویسید که اعداد بین ۱۰ تا 100 را چاپ کند، اما فقط اعدادی که به 7 ختم نمی‌شوند را چاپ کند.

8. برنامه‌ای بنویسید که جدول ضرب اعداد ۱ تا ۵ را چاپ کند، اما وقتی که به حاصل ضرب ۲۰ رسید، حلقه را متوقف کند. از حلقه تو در تو و دستور break استفاده کنید.

1: 1 2 3 4 5

2: 2 4 6 8 10

3: 3 6 9 12 15

4: 4 8 12 16

5: 5 10 15

9. برنامه‌ای بنویسید که اعداد ۱ تا ۳۰ را چاپ کند، اما اعداد مضرب ۳ را نادیده بگیرد (چاپ نکند). از دستور continue استفاده کنید

جلسه 5

1. برنامه ای بنویسید که تا زمانی که کار بر حرف q را تایپ نکرده از کاربر ورودی بگیرد و بعد در انتها ورودی های کاربر را بصورت استرینگ زیر هم چاپ کند (راهنمایی: یک راه حل ممکن استفاده از حلقه while و دستور break می باشد).

2. برنامه‌ای بنویسید که از کاربر یک عدد دریافت کند و سپس از ۱ تا ۵۰ اعداد را بررسی کند. اگر عدد ورودی را پیدا کرد، از حلقه خارج شود و پیغام «عدد پیدا شد» را چاپ کند. اگر عدد پیدا نشد، پیغام «عدد پیدا نشد» را بعد از بررسی تمام اعداد چاپ کند.

3. برنامه ای بنویسید که تمام مقادیر داخل یک لیست را در یکدیگر ضرب کند.
4. برنامه ای بنویسید که مقادیر تکراری داخل یک لیست را حذف کند.
5. برنامه ای بنویسید که 2 لیست بگیرد و مقادیر مشترک لیست ها را در یک لیست جداگانه ای خروجی دهد.
6. برنامه ای بنویسید که 2 لیست بگیرد و مقادیر غیر مشترک لیست ها را در یک لیست جداگانه ای خروجی دهد.
7. برنامه ای بنویسید که یک مقدار لیست را گرفته و تعیین کند که آیا مقادیر درون آن به ترتیب صعودی مرتب شده اند یا نه؟
8. برنامه ای بنویسید که یه جمله گرفته و تعیین کند که کدامیک از حروف صدا دار در آن متن وجود دارد. خروجی بصورت لیست برگرداند و مقدار تکراری در این لیست نباشد
9. لیستی از اعداد فرد بین ۰ تا ۵۰ ایجاد کنید در این لیست بگردید و اعدادی که بر ۵ یا ۳ بخشپذیر است را به انتهای همان لیست `append` کنید
10. لیستی به طول ۱۰۰ آیتم ایجاد کنید برای هر آیتم لیست بصورت رندم یکی رشته های زیر را انتخاب کنید. راهنمایی : برای این هدف می تونید از کتابخانه `random` و تابع `choice` استفاده کنید (آیتم های انتخابی: `hello, yello, mello, hallo`) :
11. در لیست تولید شده تمرین قبل آیتم های دارای غلط املایی را حذف کنید. (یعنی هر آیتمی که `hello` نیست)

جلسه 6

1. دو لیست زیر را در نظر بگیرید لیست اول شامل اقلام مورد نیاز برای خرید و لیست دوم شامل مقدار مورد نیاز می باشد.

```
list_1=[milk,suger,butter,yogurt,cheese]
```

```
list_2=[5,2,10,1,3]
```

برنامه ای بنویسید که این دو لیست را در یک دیکشنری ادغام کند آیتم های لیست اول کلید و آیتم های لیست دوم مقادیر متناظر باشد.

2. برنامه ای بنویسید که مقدار ماکزیمم و مینیمم `value` های دیکشنری را خروجی دهد.
3. برنامه ای بنویسید که یک جمله گرفته و تعداد تکرار هر کلمه را در یک دیکشنری ذخیره کند.
4. برنامه ای بنویسید که دو دیکشنری مانند زیر گرفته و اینها را در هم ادغام کند و برای کلیدهای یکسان در دیکشنری ها مقادیر آنها جمع کند.

```
d1 = {'a': 100, 'b': 200, 'c':300}
```

```
d2 = {'a': 300, 'b': 200, 'd':400}
```

خروجی مورد انتظار: `{'a': 400, 'b': 400, 'd': 400, 'c': 300}`

6. تابعی بنویسید که یک لیست به عنوان آرگومان ورودی گرفته و یک لیست بصورت برعکس لیست اول خروجی دهد.

7. تابعی بنویسید یک لیست بصورت ترکیبی از استزینگ و ارقام گرفته و بزرگترین مقدار لیست را خروجی دهد. (بزرگترین مقدار = استزینگ با بیشترین طول یا عددی با بیشترین مقدار)

8. یک تابعی به نام reverse بنویسید که یک عدد صحیح (مثبت یا منفی) را دریافت کند و معکوس آنرا برگرداند

input: 1234 output: 4321

input: -2456 output: -6542

جلسه 7

1. تابعی بنویسید برای کنترل سرعت خودرو: این تابع یک آرگومان میگیرد که نمایانگر سرعت است. اگر سرعت کمتر از 70 باشد مقدار "good" بر می گرداند. در غیر اینصورت به ازای هر 5 کیلومتر سرعت بیشتر از مقدار مجاز 70 یک امتیاز منفی به راننده می دهد و در آخر مجموع میزان امتیاز منفی را خروجی می دهد. مثلاً اگر سرعت 80 باشد امتیاز منفی 2 را خروجی می دهد. و اگر یک راننده بیشتر از 12 امتیاز منفی داشته باشد گواهینامه راننده باطل شده و تابع پیام "گواهینامه شما باطل شد" را پرینت می گیرد

2. یک تابعی به نام avglengh بنویسید که یک رشته حاوی یک جمله انگلیسی را دریافت کند و میانگین طول کلمات آنرا برگرداند، بدیهی است که علانم نگارشی نظیر . : ؟ ! ، کلمه محسوب نمی شوند

3. یک تابع به نام addString بنویسید که دو عدد را بصورت رشته متنی دریافت کند و حاصل جمع آنها را محاسبه و بازگرداند.

a. num1: "423" num2: "25" output: 458

4. تابعی به نام firstUChar بنویسید که یک رشته متنی را بعنوان ورودی بگیرد و اولین کاراکتر غیر تکراری آنرا یافته و اندیس آن در رشته متنی را برگرداند، در صورتیکه چنین کاراکتری را پیدا نکرد عدد منفی یک را برگرداند

5. تابعی به نام movSeifr بنویسید که یک لیست را بعنوان ورودی بگیرد و صفری که موجود در لیست را به انتهای لیست منتقل کند، بدون اینکه ترتیب سایر اعضای آرایه بهم بخورد:

a. input: [1,0,3,4,0,0,7,5,0,6]

b. output: 6[1,3,4,7,5,6,0,0,0,0]

6. تابعی به نام fillNone بنویسید که اعضای خالی در جدول را با عدد قبلی آن جایگزین کند، اولین عضو اگر خالی بود با صفر جایگزین شود:

a. input: [1,None,2,None,4,None,4,5,None]

b. output: [1,1,2,2,4,4,4,5,5]

7. لیستی از دیکشنریهای حاوی اطلاعات افراد را دریافت کنید و افرادی را که سن آنها بالای ۲۵ سال است، جدا کنید

a. ورودی نمونه:

b. people = [{"name": "Ali", "age": 23}, {"name": "Mina", "age": 30}, {"name": "Reza", "age": 25}, {"name": "Sara", "age": 28}]

c. خروجی مورد انتظار: [{"name": "Mina", "age": 30}, {"name": "Sara", "age": 28}]

8. از یک لیست تو در تو (لیستی از لیستها)، تمام اعداد منحصر به فرد را استخراج کنید و به صورت صعودی مرتب کنید

a. ورودی نمونه: `nested_list = [[1, 2, 3], [2, 3, 4], [3, 4, 5]]`

b. خروجی مورد انتظار: `[1, 2, 3, 4, 5]`

جلسه 8

1. تابعی بنویسید که شاخص کیفیت آلودگی هوا را با استفاده از یک API گرفته و در یک فایل تاریخ شمسی و شاخص آلودگی را ذخیره کند.

2. تابعی بنویسید که n خط اول یک فایل را بخواند. (n آرگومان ورودی تابع است)

3. تابعی به نام `findM` بنویسید که دو تا فایل را بصورت ورودی دریافت کند و تعداد کلمات مشترک آنها را برگرداند

4. تابعی بنویسید به نام `copy_file` که دو نام فایل را به عنوان ورودی دریافت کند و محتوای فایل اول را به فایل دوم کپی کند.

5. تابعی بنویسید به نام `remove_empty_lines` که نام یک فایل را به عنوان ورودی دریافت کند و تمام خطوط خالی آن فایل را حذف کند. سپس فایل جدید را با نام مشابه ذخیره کند.

6. تابعی بنویسید به نام `search_word` که نام یک فایل و یک کلمه را به عنوان ورودی دریافت کند و تعداد تکرار آن کلمه در فایل را برگرداند.

7. تابعی بنویسید به نام `compare_files` که دو نام فایل را به عنوان ورودی دریافت کند و تعیین کند که آیا محتوای دو فایل دقیقاً برابر هستند یا خیر.

8. تابعی بنویسید که n خط آخر یک فایل را بخواند

جلسه 11

یک برنامه مدیریت فروشگاه برای یک بستنی فروشی بنویسید.

در این فروشگاه انواع مختلفی از بستنی ها بفروش می رسد. قوانین فروشگاه و امکانات مورد نیاز نرم افزار مدیریت فروشگاه بدین شرح است:

۱- بیشتر از موجودی انبار نمی تواند بستنی بفروشد

۲- در صورت خرید بستنی به میزان بستنی خریداری شده از هر نوع به موجودی انبار اضافه می گردد.

۳- انبار ظرفیت محدودی دارد و کلاً گنجایش ۲۰۰ کیلو بستنی را دارد.

۴- انواع بستنی های موجود در این فروشگاه عبارتند از:

۴-۱- بستنی شکلاتی وزن هر بستنی ۱۰۰ گرم . قیمت ۱۰

۴-۲- بستنی میوه ای وزن هر بستنی ۵۵ گرم قیمت ۷

۴-۳- بستنی موزیکال وزن هر بستنی ۲۰۰ گرم طبق قانون فروشگاه بستنی موزیکال فقط بصورت جفتی بفروش می رسد یعنی به ازای هر سفارش از این نوع بستنی دو تا بستنی به مشتری تحویل داده می شود. و برای هر بستنی موزیکال آهنگ مخصوصی ذخیره شده است و اگر مشتری درخواست کند نرم افزار آهنگ مخصوص بستنی فروخته شده را پلی می کند . قیمت هر بستنی ۱۵ .

۴-۴- بستنی یخی وزن هر بستنی ۴۰ گرم قیمت ۳

۵- سود بستنی فروش ۲۰ درصد قیمت فروش است

قیمت ها به هزار تومان است

جلسه 13

اسکرپ کردن سایت آرشیو سایت ایران جیب مطابق با توضیحات کلاس

جلسه 14

اسکرپ کردن سایت ایران فایل و استخراج اطلاعات خانه ها و ذخیره در دیتابیس