**تمارین عملی سر کلاس**

### ماشین حساب هوشمند برای بودجه‌بندی ماهانه

**تمرین اول :‌**

برنامه‌ای بنویسید که به کاربر کمک کند بودجه ماهانه خود را بررسی کند. اطلاعات زیر را از کاربر بگیرد:

* مبلغ درآمد ماهانه
* هزینه‌های زیر به‌صورت جداگانه خوراک، اجاره، حمل و نقل، تفریح

برنامه باید:

1. جمع کل هزینه‌ها را محاسبه و از درآمد ماهانه کسر کند.
2. اگر مجموع هزینه‌ها از درآمد ماهانه بیشتر باشد، پیام "بودجه شما کافی نیست!" نمایش دهد.
3. اگر هزینه تفریح بیشتر از ۳۰٪ کل درآمد باشد، هشدار دهد که "هزینه تفریح بیش از حد است."
4. در پایان، میزان باقیمانده بودجه را به همراه واحد پولی چاپ کند.

**تمرین ۲: برنامه حساب کاربری فروشگاه**

برنامه‌ای بنویسید که از کاربر مشخصات یک خرید آنلاین دریافت کند:

* نام کالا (به‌صورت استرینگ)
* تعداد کالا (به‌صورت عدد صحیح)
* قیمت واحد کالا (به‌صورت عدد اعشاری)
* کد تخفیف : اگر کاربر کد خاصی مانند "DISCOUNT10" وارد کرد، ۱۰٪ تخفیف اعمال شود

برنامه باید:

1. قیمت نهایی را با توجه به تعداد و قیمت واحد محاسبه کند.
2. اگر کاربر کد تخفیف وارد کرده باشد، تخفیف را اعمال کرده و قیمت نهایی را محاسبه و نمایش دهد.
3. اگر قیمت نهایی بیش از ۱۰۰ هزار تومان باشد، پیامی نمایش دهد که "شما مشمول ارسال رایگان هستید."

**تمرین when**

برنامه‌ای بنویسید که وضعیت آب‌و‌هوا رو از کاربر بگیره و یه فعالیت پیشنهادی برای اون روز بهش بده.

کاربر یکی از این وضعیت‌ها رو وارد می‌کنه: "آفتابی"، "بارانی"، "برفی"، "ابری"

برنامه با توجه به وضعیت آب‌و‌هوا یکی از پیشنهادهای زیر رو بهش می‌ده:

* آفتابی: "پیاده‌روی در پارک رو امتحان کن!"
* بارانی: "کتاب بخون یا فیلم ببین."
* برفی: "یک آدم برفی بساز!"
* ابری: "با دوستات برو کافه."
* بادی: "بادبادک هوا کن!"

اگه کاربر یه چیز دیگه وارد کرد، برنامه باید پیام بده: "وضعیت آب‌وهوا نامعتبره!"

**تمارین بعد از لیست:**

### تمرین جمع‌آوری اطلاعات با لیست

برنامه‌ای بنویسند که اسامی پنج دوست کاربر را از ورودی دریافت کرده و در یک لیست ذخیره کند. سپس همه‌ی اسامی را به ترتیب نمایش دهد.

### تمرین وارونه کردن لیست (Reversed())

برنامه‌ای بنویسند که ۵ عدد از کاربر بگیرد و آنها را در یک لیست ذخیره کند. سپس لیست را به صورت وارونه (از آخر به اول) چاپ کند.

### برنامه تحلیل کلمات

برنامه‌ای بنویسید که از کاربر بخواهد **چندین کلمه** وارد کند. دریافت کلمات باید تا زمانی که کاربر عبارت "پایان" را وارد نکرده، ادامه یابد. سپس برنامه موارد زیر را نمایش دهد:

1. **تعداد کل کلمات** وارد شده.
2. **بلندترین کلمه** و **کوتاه‌ترین کلمه** وارد شده.

### تمرین جستجو در لیست (contains)

برنامه‌ای بنویسند که از کاربر چند اسم دریافت کند و آنها را در یک لیست ذخیره کند. سپس از کاربر یک اسم بگیرد و بررسی کند که آیا این اسم در لیست وجود دارد یا خیر.

**تمارین بعد از حلقه :**

**✴️ تمرین دوم کاتلین: پیدا کردن عدد بزرگتر (while)**

برنامه‌ای بنویسین که از کاربر **۴ عدد** بگیره (با readLine()?.toInt() یا مشابه) و با استفاده از **حلقه while** عدد بزرگ‌تر رو بین اون‌ها پیدا کنه و چاپ کنه.

**شمارنده اعداد و محاسبه جمع**

✅ قدم اول:  
یه عدد از کاربر بگیرین.

✅ قدم دوم:  
با استفاده از یک حلقه، تمام **اعداد زوج و فرد تا اون عدد** رو به‌صورت جداگانه چاپ کنین.

✅ قدم سوم:  
در نهایت، جمع اعداد زوج و جمع اعداد فرد رو هم جداگانه محاسبه کنین و نمایش بدین.

📌 می‌تونین از for یا while استفاده کنین.

**✴️ تمرین چهارم کاتلین: بررسی عدد اول بودن**

✅ قدم اول:  
از کاربر یک عدد بگیرین و چک کنین که آیا عدد **اول** هست یا نه.

✅ نکته:  
عدد اول فقط بر خودش و ۱ بخش‌پذیره. پس باید بررسی کنین که آیا عدد مورد نظر بر عدد دیگه‌ای بخش‌پذیر هست یا نه.

✅ مثال:  
اگر کاربر ۷ وارد کرد:

عدد ۷ اول است

اگر ۱۰ وارد کرد:

عدد ۱۰ اول نیست

**ماشین حساب ساده + مدیریت خطا**

✅ قدم اول:  
از کاربر بخواین **دو عدد** و یک عملیات (مثل add, subtract, multiply, divide) وارد کنه.

✅ قدم دوم:  
عملیات انتخاب‌شده رو اجرا کنین و نتیجه رو چاپ کنین.

✅ مدیریت خطاها:

* اگر کاربر به‌جای عدد، چیز نامعتبری وارد کرد، یه پیام خطای مناسب نشون بدین (NumberFormatException).
* اگر عملیات تقسیم انجام شد و عدد دوم صفر بود، پیام تقسیم بر صفر مجاز نیست رو نمایش بدین.
* حتی اگه خطا رخ داد، برنامه نباید بسته شه — دوباره از کاربر ورودی جدید بگیرین.

**تمارین جامع فصل اول**

**بازی حدس عدد (Number Guessing Game)**

**هدف:** تمرین با متغیرها، عملگرها، عبارات شرطی، و حلقه‌ها.

**شرح پروژه:** یک بازی طراحی کنید که در آن برنامه یک عدد تصادفی بین ۱ تا ۱۰۰ انتخاب کند و کاربر باید آن عدد را حدس بزند. برنامه باید به کاربر بگوید که آیا عدد ورودی‌اش بزرگ‌تر، کوچک‌تر یا برابر با عدد مخفی است.

1. برنامه باید یک عدد تصادفی بین ۱ تا ۱۰۰ انتخاب کند (با استفاده از Random).
2. از کاربر بخواهید که یک عدد وارد کند.
3. برنامه باید در هر حدس به کاربر بگوید که عددش "بزرگتر" است، "کوچکتر" است، یا درست است.
4. اگر کاربر عدد درست را حدس زد، پیام تبریک نمایش داده شود و برنامه پایان یابد.
5. در غیر این صورت، کاربر می‌تواند به حدس‌های خود ادامه دهد.
6. از حلقه while برای تکرار حدس‌ها استفاده کنید و بازی را تا زمانی که کاربر عدد درست را پیدا کند ادامه دهید.

**راهنمایی:**

* از when برای مقایسه ورودی کاربر با عدد مخفی استفاده کنید.
* برای تولید عدد تصادفی از Random().nextInt(1, 101) استفاده کنید.

**بررسی پسورد قوی (Password Strength Checker)**

**هدف:** استفاده از استرینگ، عبارات شرطی، و حلقه‌ها.

**شرح پروژه:** یک برنامه بنویسید که قدرت یک پسورد را بررسی کند. پسورد ورودی از کاربر دریافت می‌شود و باید شرایط زیر را بررسی کند:

1. حداقل ۸ کاراکتر طول داشته باشد.
2. حداقل یک حرف بزرگ، یک حرف کوچک، یک عدد، و یک کاراکتر خاص (@, #, $, ! و ...) داشته باشد.
3. برنامه به کاربر بگوید که آیا پسوردش قوی است یا ضعیف.

**راهنمایی:**

* از عبارات شرطی (if و else) برای بررسی شرایط استفاده کنید.
* از حلقه for برای پیمایش کاراکترهای پسورد استفاده کنید.

**تمرین تابع**

**برنامه تبدیل واحد**

✅ قدم اول:  
از کاربر **دو تا ورودی بگیرین**:

1. یه عدد (مثلاً: 5)
2. نوع واحد (مثلاً: "متر" یا "کیلوگرم" یا "ثانیه")

✅ قدم دوم:  
یه تابع به اسم convert بنویسین که این ورودی‌ها رو بگیره و مقدار رو به واحد دیگه تبدیل کنه.

📌 چند نمونه تبدیل که می‌تونین پیاده‌سازی کنین:

* **متر** به **سانتی‌متر** و بالعکس
* **کیلوگرم** به **گرم** و بالعکس
* **ثانیه** به **دقیقه** و **ساعت**

✅ قدم آخر:  
مقدار تبدیل‌شده رو توی کنسول نمایش بدین.

👀 با when یا if می‌تونین تعیین کنین که کدوم نوع تبدیل انجام بشه.

**گیم شانس (حدس عدد)**

برنامه‌ای بنویسید که: یک عدد تصادفی بین ۱ تا ۱۰۰ تولید کند. کاربر سعی کند این عدد را حدس بزند. یک تابع به نام checkGuess بنویسید که عدد ورودی را با عدد تصادفی مقایسه کرده و راهنمایی "بالاتر" یا "پایین‌تر" را به کاربر بدهد.

تعداد تلاش‌های کاربر را شمارش کرده و در انتها نمایش دهید.

لامبدا

### ****بررسی پالیندروم با Lambda و Higher-Order Function****

### تابعی به نام checkStringبنویسید که یک رشته و یک تابع Lambda به عنوان ورودی بگیرد و بررسی کند آیا شرط مورد نظر روی رشته اعمال می‌شود یا نه. سپس از این تابع برای بررسی پالیندروم بودن یک رشته استفاده کنید.

### کلمه های پالیندورم مثل : رادار، مادام و ...

### راهنمایی : امضای تابع :

### fun checkString(input: String, condition: (String) -> Boolean): Boolean