

---

---

# پیاده‌سازی پروژه- های برنامه‌نویسی با پایتون

---

---

**تألیف:**

دکتر رمضان عباس نژادورزی  
دکتر حمیدرضا حسن نژاد مرزونی



فن آوری نوین

---

---

سرشناسه	: عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	: پیاده‌سازی پروژه‌های برنامه‌نویسی با پایتون / تألیف رمضان عباس نژاد ورزی، حمیدرضا حسن نژاد مرزونی.
مشخصات نشر	: بابل: فناوری نوین، ۱۴۰۰.
مشخصات ظاهری	: ۳۱۹ص: مصور(رنگی)، جدول.
شابک	: ۹۷۸-۶۲۲-۷۳۹۳-۴۶-۰۰
وضعیت فهرست نویسی	: فیپا
یادداشت	: کتابنامه: ص: ۳۱۹.
موضوع	: پایتون (زبان برنامه‌نویسی کامپیوتر)
موضوع	: Python (Computer program language)
شناسه افزوده	: حسن نژاد مرزونی، حمیدرضا، ۱۳۶۶ -
رده بندی کنگره	: ۷۶/۷۳QA
رده بندی دیویی	: ۰۰۵/۱۳۳
شماره کتابشناسی ملی	: ۸۴۴۵۵۲۲
اطلاعات رکورد کتابشناسی	: فیپا

@fanavarienovinpub

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

### پیاده‌سازی پروژه‌های برنامه‌نویسی با پایتون

تألیف: رمضان عباس نژاد ورزی، حمیدرضا حسن نژاد مرزونی

نوبت چاپ: چاپ اول

سال چاپ: تابستان ۱۴۰۰

شمارگان: ۲۰۰

قیمت: ۹۵۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک چاپی: ۹۷۸-۶۲۲-۷۳۹۳-۴۶-۰۰

شابک الکترونیکی: ۹۷۸-۶۲۲-۷۳۹۳-۴۵-۰۳

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

**فروشگاه و پخش کتاب چاپی: تهران، تلفن ۶۶۴۰۰۲۲۰-۶۶۴۰۰۱۴۴**

تهران، خ اردیبهشت، نش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

## فهرست مطالب

<b>فصل اول: پیاده‌سازی پروژه‌های ساده و کوچک</b>	<b>۷</b>
۱-۱. پیاده‌سازی برنامه حدس زدن عدد	۷
۱-۲. پیاده‌سازی برنامه بازی سنگ، کاغذ و قیچی	۹
۱-۳. پیاده‌سازی برنامه ساخت ماشین حساب ساده	۱۲
۱-۴. پیاده‌سازی برنامه جمع و تفریق اعداد بزرگ	۱۵
۱-۵. پیاده‌سازی برنامه دریافت روزی از سال، نمایش ماه و فصل	۱۸
۱-۶. پیاده‌سازی برنامه محاسبه مقدار یک سری پیچیده با حلقه تکرار	۲۰
۱-۷. پیاده‌سازی برنامه نمایش جفت اعداد متحابه	۲۱
۱-۸. پیاده‌سازی برنامه انجام عملیات مختلف بر روی آرایه با توابع	۲۱
۱-۹. پیاده‌سازی برنامه انجام عملیاتی از قبیل بالا مثلثی، پایین مثلثی و غیره بر روی آرایه با توابع	۲۷
۱-۱۰. پیاده‌سازی برنامه انجام عملیاتی از قبیل +، -، * بر روی آرایه با numpy	۳۰
۱-۱۱. پیاده‌سازی برنامه تعیین درصد حروف، ارقام، حروف کوچک، حروف بزرگ، حروف صدادار در یک رشته ورودی	۳۲
۱-۱۲. پیاده‌سازی برنامه پیاده‌سازی عملگرهای مختلف از قبیل +، -، * و غیره برای کلاس کسر	۳۳
۱-۱۳. پیاده‌سازی برنامه کلاس‌های نقطه و دایره از طریق وراثت	۳۵
۱-۱۴. پیاده‌سازی برنامه دریافت سال و نمایش تقویم آن سال	۳۷
۱-۱۵. پیاده‌سازی برنامه تولید رنگ تصادفی در Tkinter	۴۰
۱-۱۶. پیاده‌سازی برنامه نمایش تکرار کلمات یک فایل به صورت نمودار ستونی	۴۲
۱-۱۷. پیاده‌سازی برنامه رسم گل آفتابگردان در turtle	۴۴
۱-۱۸. پیاده‌سازی برنامه رسم قیف در turtle	۴۶
۱-۱۹. پیاده‌سازی سه عمل جمع، تفریق و توان با Kivy	۴۷
۱-۲۰. پیاده‌سازی سه عمل تفریق، تقسیم و باقی‌مانده تقسیم با فایبل kv	۴۹
<b>فصل دوم: پیاده‌سازی پروژه‌های متوسط</b>	<b>۵۴</b>
۲-۱. پیاده‌سازی برنامه محاسبه مقدار چند سری پیچیده با توابع	۵۴
۲-۲. پیاده‌سازی برنامه پیاده‌سازی توابعی از قبیل Lower، Upper، strLen و غیره برای رشته‌ها	۵۶
۲-۳. پیاده‌سازی برنامه محاسبه سن کاربر	۶۱
۲-۴. پیاده‌سازی برنامه محاسبه حقوق یک هفته کارمند با Tkinter	۶۵
۲-۵. پیاده‌سازی برنامه تولیدکننده کلمه عبور با Tkinter	۷۳
۲-۶. پیاده‌سازی برنامه ذخیره اطلاعات کتاب با Tkinter و SQLite	۷۸
۲-۷. پیاده‌سازی برنامه بازی مار (Snake) با pygame	۸۷
۲-۸. پیاده‌سازی برنامه بازی دوز (Tic Tac Toe) با pygame	۹۳
۲-۹. پیاده‌سازی برنامه انتخاب فونت و دادن جلوه ویژه به متن در PyQT	۱۰۱
۲-۱۰. پیاده‌سازی برنامه کشیدن و رها کردن در PyQT	۱۰۵
۲-۱۱. پیاده‌سازی برنامه رسم درخت در turtle	۱۰۸
۲-۱۲. پیاده‌سازی برنامه نمایش تعداد روزهای مانده یک پروژه در Tkinter	۱۰۹
۲-۱۳. پیاده‌سازی برنامه نوشتن متون با کلیک ماوس و کشیدن آن در Tkinter	۱۱۰
۲-۱۴. پیاده‌سازی برنامه هشداردهنده در Tkinter	۱۱۲

۱۱۳	..... PyQT در LED به صورت Slider به نمایش مقدار
۱۱۵	..... SMTP. پیاده سازی برنامه ارسال پست الکترونیکی با
۱۱۷	..... ۲- ۱۷. پیاده سازی برنامه ترسیم دو چهره انسان و حرکت دادن آن ها در مقابل یکدیگر
۱۲۰	..... ۲- ۱۸. پیاده سازی برنامه یک ماشین حساب پیشرفته با Tkinter
۱۳۰	..... ۲- ۱۹. پیاده سازی ماشین حساب ساده با Kivy
۱۳۲	..... ۲- ۲۰. پیاده سازی برنامه شبیه ساز کرومتر در کیوی
۱۳۷	..... ۲- ۲۱. پیاده سازی برنامه انتخاب کننده فایل در کیوی
۱۳۹	..... ۲- ۲۲. پیاده سازی بازی Pong در کیوی
۱۴۳	..... ۲- ۲۳. پیاده سازی برنامه رسم اشکال گرافیکی مختلف در کیوی
۱۵۱	..... ۲- ۲۴. پیاده سازی برنامه انتخاب مقدار رنگها با Slider ها و ترکیب رنگها در کیوی
۱۵۳	..... ۲- ۲۵. پیاده سازی برنامه ایجاد نوار پیشرفت دایره ای در کیوی
۱۵۷	<b>..... فصل سوم: پیاده سازی پروژه های پیچیده و بزرگ</b>
۱۵۷	..... ۳- ۱. پیاده سازی برنامه آزمایش عملکرد موتور گرافیکی کیوی
۱۵۹	..... ۳- ۲. پیاده سازی برنامه رسم دایره با تعیین زاویه شروع، پایان با Slider ها در Kivy
۱۶۴	..... ۳- ۳. پیاده سازی برنامه رسم خط با انتخاب ویژگی های آن به همراه اعمال انیمیشن به خط در کیوی
۱۷۵	..... ۳- ۴. پیاده سازی برنامه نوشتن متن از طریق ماوس و Touch در کیوی
۱۸۰	..... ۳- ۵. پیاده سازی برنامه تبدیل مبناهای مختلف به یکدیگر با Tkinter
۱۹۹	..... ۳- ۶. پیاده سازی برنامه ویراستار Notepad با Tkinter
۲۱۵	..... ۳- ۷. پیاده سازی برنامه نمایش LED ها در Tkinter
۲۲۲	..... ۳- ۸. پیاده سازی دفترچه تلفن با SQLite و Tkinter
۲۲۸	..... ۳- ۹. پیاده سازی برنامه ذخیره و بازیابی اطلاعات یک شخص خاص در بانک اطلاعاتی MySQL با Tkinter
۲۴۶	..... ۳- ۱۰. پیاده سازی برنامه نگهداری و بازیابی اطلاعات افراد با MySQL و Tkinter
۲۶۲	..... ۳- ۱۱. پیاده سازی برنامه ذخیره و بازیابی اطلاعات کاربر در بانک اطلاعاتی با کیوی
۲۷۶	..... ۳- ۱۲. پیاده سازی برنامه بانک اطلاعاتی فروشگاه ساده با PyQT
۳۱۹	<b>..... منابع:</b>

## مقدمه

پایتون یک زبان برنامه‌نویسی همه منظوره، سطح بالا، شیء‌گرا و مفسری است که توسط فان روسوم در سال ۱۹۹۱ در کشور هلند طراحی گردید. فلسفه ایجاد این زبان بر دو هدف اصلی خوانایی بالایی برنامه‌ها نوشته شده، کوتاهی و بازدهی نسبی بالای آن است. کلمات کلیدی این زبان به صورت حداقلی تهیه شده‌اند و در مقابل کتابخانه‌هایی که در اختیار کاربر است، بسیار وسیع هستند. تاکنون کتاب‌های منتشر شده به بیان مفاهیم برنامه‌نویسی با حل مسائل ساده اقدام می‌نمودند که با مطالعه این کتاب‌ها برنامه‌نویسان در نوشتن پروژه‌های واقعی با مشکل مواجهه می‌شدند. اما کتاب حاضر اولین کتاب حل پروژه‌های برنامه‌نویسی پایتون است که با پیاده‌سازی پروژه‌های واقعی برنامه‌نویسان را برای حل پروژه‌های واقعی آماده می‌کند. این کتاب شامل سه فصل است که عبارت انداز:

فصل اول، پیاده‌سازی پروژه‌های ساده و کوچک را با ۲۰ پروژه آموزش می‌دهد.

فصل دوم، پیاده‌سازی پروژه‌های متوسط را با ۲۵ پروژه آموزش می‌دهد.

فصل سوم، پیاده‌سازی پروژه‌های پیچیده و بزرگ را با ۱۲ پروژه آموزش می‌دهد.

در این کتاب برای پیاده‌سازی پروژه‌ها از کتابخانه‌هایی از قبیل `pyqt5`, `tkinter`, `matplotlib`, `pygame`, `partial`, `graphics`, `image` استفاده شده است. برخی از این کتابخانه‌ها ممکن است در سیستم‌تان نصب نباشند. پس ممکن است در اجرای بعضی از پروژه‌ها با خطا مواجهه شوید. برای رفع این نوع خطا باید بسته موردنظرتان را بر روی کامپیوترتان نصب نمایید. برای نصب یک مازول باید به خط فرمان (cmd) بروید و یکی از دستورات زیر را تایپ کنید:

**نام بسته pip install**

**نام بسته pip3 install**

به‌عنوان مثال، برای نصب بسته kivy، یکی از دستورات زیر را تایپ کنید:

**1. pip install kivy**

**2. pip3 install kivy**

در ضمن برای آموزش مفاهیم پایتون می‌توانید به کتاب‌های زیر از همین انتشارات مراجعه فرمایید:

۱. کتاب مرجع کامل برنامه‌نویسی

۲. کتاب ساختمان داده‌ها با پایتون

۳. طراحی واسط کاربر با پایتون (Tkinter)

۴. آموزش گام به گام بانک اطلاعاتی با پایتون

۵. طراحی رابط کاربری با PYQT در پایتون

۶. آموزش گام به گام برنامه‌نویسی گرافیک در پایتون

۷. آموزش گام به گام برنامه‌نویسی کیوی

از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و

انتقادات را به آدرس پست الکترونیک [fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com) ارسال نمایند.

در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

[fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com)

مؤلفین - تابستان ۱۴۰۰

## برنامه نویسی پایتون مرجع کامل

### برنامه نویسی پایتون

تألیف:  
 دکتر جواد وحیدی  
 عضو هیات علمی دانشکده علوم و صنعت ایران  
 دکتر رمضان عباس نژاد ووزی

سری توسعه دهندگان دایبل

## پایتون

### برای برنامه نویسان

با مثالها و تکنیکهای هوش مصنوعی

برخی از عناوین مهم:  
 پایتون برای مبتدیان  
 مفاهیم پایه پایتون  
 کار با داده‌ها و فایل‌ها  
 کار با پایگاه داده‌ها  
 کار با شبکه‌ها  
 کار با اینترنت  
 کار با APIها  
 کار با کتابخانه‌های علمی  
 کار با کتابخانه‌های مالی  
 کار با کتابخانه‌های بازی  
 کار با کتابخانه‌های تصویربرداری  
 کار با کتابخانه‌های پردازش تصویر  
 کار با کتابخانه‌های پردازش صوت  
 کار با کتابخانه‌های پردازش زبان طبیعی  
 کار با کتابخانه‌های پردازش داده‌های بزرگ  
 کار با کتابخانه‌های پردازش داده‌های حجیم  
 کار با کتابخانه‌های پردازش داده‌های توزیع شده  
 کار با کتابخانه‌های پردازش داده‌های ابری  
 کار با کتابخانه‌های پردازش داده‌های بیگ دیتا  
 کار با کتابخانه‌های پردازش داده‌های کلان داده  
 کار با کتابخانه‌های پردازش داده‌های کلان داده ابری  
 کار با کتابخانه‌های پردازش داده‌های کلان داده ابری توزیع شده  
 کار با کتابخانه‌های پردازش داده‌های کلان داده ابری توزیع شده بیگ دیتا  
 کار با کتابخانه‌های پردازش داده‌های کلان داده ابری توزیع شده بیگ دیتا ابری  
 کار با کتابخانه‌های پردازش داده‌های کلان داده ابری توزیع شده بیگ دیتا ابری توزیع شده

مؤلفین: دکتر رمضان عباس نژاد ووزی  
 دکتر جواد وحیدی

## مبانی کامپیوتر و برنامه سازی با رویکرد الگوریتم و فلوچارت

### پایه سازی شده با پایتون

مؤلفین: دکتر رمضان عباس نژاد ووزی  
 دکتر جواد وحیدی  
 دکتر محمد احمدزاده

برخی از عناوین مهم:  
 مبانی کامپیوتر و سیستم‌های رایانه  
 مبانی سیستم‌های کامپیوتری  
 مبانی سیستم‌های اطلاعاتی  
 مبانی سیستم‌های داده‌های بزرگ  
 مبانی سیستم‌های داده‌های کلان داده  
 مبانی سیستم‌های داده‌های کلان داده ابری  
 مبانی سیستم‌های داده‌های کلان داده ابری توزیع شده  
 مبانی سیستم‌های داده‌های کلان داده ابری توزیع شده بیگ دیتا  
 مبانی سیستم‌های داده‌های کلان داده ابری توزیع شده بیگ دیتا ابری  
 مبانی سیستم‌های داده‌های کلان داده ابری توزیع شده بیگ دیتا ابری توزیع شده

## برنامه نویسی کیوی به زبان ساده

### برنامه نویسی اندروید با پایتون

مؤلفین: دکتر رمضان عباس نژاد ووزی  
 دکتر محمد احمدزاده

برخی از عناوین مهم:  
 مبانی برنامه نویسی کیوی  
 مبانی برنامه نویسی اندروید  
 مبانی برنامه نویسی اندروید با پایتون  
 مبانی برنامه نویسی اندروید با پایتون و کیوی  
 مبانی برنامه نویسی اندروید با پایتون و کیوی و اندروید

## آموزش گام به گام بانک اطلاعاتی با پایتون

مؤلف: دکتر رمضان عباس نژاد ووزی

برخی از عناوین مهم:  
 مبانی بانک اطلاعاتی  
 مبانی بانک اطلاعاتی با پایتون  
 مبانی بانک اطلاعاتی با پایتون و کیوی  
 مبانی بانک اطلاعاتی با پایتون و کیوی و اندروید

## حل مسائل پایتون (حل ۶۵۰ برنامه - مرجع کامل)

مؤلفین: دکتر رمضان عباس نژاد ووزی  
 دکتر محمد احمدزاده

برخی از عناوین مهم:  
 مبانی حل مسائل پایتون  
 مبانی حل مسائل پایتون با پایتون  
 مبانی حل مسائل پایتون با پایتون و کیوی  
 مبانی حل مسائل پایتون با پایتون و کیوی و اندروید

## یادگیری عمیق برای انسانها

### درک کنید که شبکه‌های عصبی عمیق چگونه کار می‌کنند

مؤلف: محمّد شایسته  
 مترجمین: دکتر جواد وحیدی، حسن نژاد ووزی  
 دکتر محمد احمدزاده

برخی از عناوین مهم:  
 مبانی یادگیری عمیق  
 مبانی یادگیری عمیق برای انسانها  
 مبانی یادگیری عمیق برای انسانها و ماشینها  
 مبانی یادگیری عمیق برای انسانها و ماشینها و اندروید

## طراحی رابط کاربری در پایتون

### PyQt در پایتون

مؤلف: دکتر رمضان عباس نژاد ووزی  
 دکتر محمد احمدزاده

برخی از عناوین مهم:  
 مبانی طراحی رابط کاربری  
 مبانی طراحی رابط کاربری در پایتون  
 مبانی طراحی رابط کاربری در پایتون و کیوی  
 مبانی طراحی رابط کاربری در پایتون و کیوی و اندروید

## ساختمان داده با پایتون

مؤلفین: دکتر جواد وحیدی، محمد عباس علی، شایسته، محمّد احمدزاده  
 دکتر رمضان عباس نژاد ووزی

برخی از عناوین مهم:  
 مبانی ساختمان داده  
 مبانی ساختمان داده با پایتون  
 مبانی ساختمان داده با پایتون و کیوی  
 مبانی ساختمان داده با پایتون و کیوی و اندروید

## پروژه‌های برنامه‌نویسی ساده و کوچک

## ۱-۱. پیاده‌سازی برنامه حدس زدن عدد

برنامه‌ای که یک عدد تصادفی از ۱ تا ۱۰ ایجاد می‌کند. پس از ایجاد عدد توسط کامپیوتر کاربر باید آن را حدس بزند. هر بار که کاربر عدد را اشتباه حدس بزند، کاربر را راهنمایی کرده، تا حدس زدن عدد را برای او آسان کند (البته با کاهش امتیاز او).

۱. یک فایل جدید پایتون ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```
1 import random
2 attempts_list = []
3 def show_score():
4     if len(attempts_list) <= 0:
5         print("There is currently no high score, it's yours for the
6             taking!")
7     else:
8         print("The current high score is {}
9             attempts".format(min(attempts_list)))
10 def start_game():
11     random_number = int(random.randint(1, 10))
12     print("Hello traveler! Welcome to the game of guesses!")
13     player_name = input("What is your name? ")
14     wanna_play = input("Hi, {}, would you like to play the guessing
15         game? (Enter Yes/No) ".format(player_name))
16     # Where the show_score function USED to be
17     attempts = 0
18     show_score()
19     while wanna_play.lower() == "yes":
20         try:
21             guess = input("Pick a number between 1 and 10: ")
22             if int(guess) < 1 or int(guess) > 10:
23                 raise ValueError("Please guess a number within the
24                     given range")
25             if int(guess) == random_number:
26                 print("Nice! You got it!")
27                 attempts += 1
28                 attempts_list.append(attempts)
29                 print("It took you {} attempts".format(attempts))
30                 play_again = input("Would you like to play again?
31                     (Enter Yes/No) ")
32                 attempts = 0
33                 show_score()
34                 random_number = int(random.randint(1, 10))
35                 if play_again.lower() == "no":
36                     print("That's cool, have a good one!")
37                     break
38                 elif int(guess) > random_number:
```

```

39         print("It's lower")
40         attempts += 1
41         elif int(guess) < random_number:
42             print("It's higher")
43             attempts += 1
44         except ValueError as err:
45             print("Oh no!, that is not a valid value. Try again...")
46             print("{}".format(err))
47     else:
48         print("That's cool, have a good one!")
49 if __name__ == '__main__':
50     start_game()

```

دستور ۱، بسته random را به برنامه اضافه می‌کند تا بتواند یک عدد تصادفی تولید کند، دستور ۲، متغیر attempts\_list را از نوع لیست با مقدار خالی ایجاد می‌کند، دستورات ۳ تا ۹، تابع show\_score() را تعریف می‌نمایند که امتیاز بازیکن را نمایش می‌دهد، دستورات ۱۰ تا ۴۸، تابع start\_game() را پیاده‌سازی می‌کنند که در این تابع اعمال زیر انجام می‌شود:

۱. یک عدد تصادفی بین ۱ تا ۱۰ تولید کرده، در متغیر random\_number قرار می‌دهد (دستور ۱۱).

۲. پیغام خوشامدگویی را نمایش می‌دهد (دستور ۱۲).

۳. نام کاربر را می‌خواند (دستور ۱۳).

۴. از کاربر می‌پرسد که آیا بازی را ادامه می‌دهد (دستورات ۱۴ و ۱۵).

۵. تعداد تلاش (attempts) را برابر ۰ قرار می‌دهد (دستور ۱۷).

۶. تابع show\_score() را فراخوانی می‌کند (دستور ۱۸).

۷. اگر کاربر "yes" را وارد کرده باشد، در یک حلقه while اعمال زیر را انجام می‌دهد:

❖ یک عدد را از کاربر دریافت می‌کند (دستور ۲۱).

❖ اگر عدد ورودی بزرگ‌تر از ۱۰ یا کوچک‌تر از ۰ باشد، یک پیغام خطا را نمایش می‌دهد (دستورات

۲۱ تا ۲۳).

❖ اگر عدد ورودی برابر random\_number باشد (دستور ۲۵)، ابتدا پیغامی را نمایش می‌دهد (دستور ۲۶)،

به تعداد تلاش یک واحد اضافه می‌کند، و تعداد تلاش را به لیست کلاس‌ها اضافه می‌کند (دستور ۲۸) و با

پیغامی تعداد تلاش را نمایش می‌دهد (دستور ۲۹)، از کاربر می‌پرسد که ادامه می‌دهد (دستورات ۳۰ و ۳۱)،

تعداد تلاش را برابر ۰ می‌کند (دستور ۳۲)، تابع show\_score() را فراخوانی می‌نماید (دستور ۳۳)، یک عدد

پروژه‌های برنامه‌نویسی ساده و کوچک ۹

تصادفی بین ۱ تا ۱۰ تولید می‌کند (دستور ۳۴)، اگر کاربر "no" را وارد کرده باشد (دستور ۳۵)، یک پیغام را نمایش داده و از حلقه خارج می‌شود (دستورات ۳۶ و ۳۷).

❖ وگرنه اگر عدد وارد شده بزرگ‌تر از عدد تصادفی باشد (دستور ۳۸)، پیغام "it's is lower" را نمایش داده، به attempts یک واحد اضافه می‌کند (دستورات ۳۹ و ۴۰).

❖ اگر عدد وارد شده کم‌تر از عدد تصادفی باشد (دستور ۱۴۰)، پیغام "it's is higher" را نمایش می‌دهد و به attempts یک واحد اضافه می‌کند (دستورات ۴۲ و ۴۳).

❖ در صورت رخ دادن خطایی در این تابع (دستور ۴۴)، پیغام "Oh no!, that is not a valid value. Try again..." را نمایش می‌دهد (دستورات ۴۵ و ۴۶).

دستور ۵۰، تابع start\_game() را فراخوانی می‌کند تا بازی شروع شود.

۲. پروژه را اجرا کرده، ورودی‌های خواسته‌شده را وارد کنید (مانند زیر) و خروجی را مشاهده نمایید:

```
Hello traveler! Welcome to the game of guesses!
What is your name? Ramzan
Hi, Ramzan, would you like to play the guessing game? (Enter Yes/No) Yes
There is currently no high score, it's yours for the taking!
Pick a number between 1 and 10 5
It's higher
Pick a number between 1 and 10 8
It's higher
Pick a number between 1 and 10 9
Nice! You got it!
It took you 3 attempts
Would you like to play again? (Enter Yes/No) No
The current high score is 3 attempts
That's cool, have a good one!
```

## ۱-۲. پیاده‌سازی برنامه بازی سنگ، کاغذ و قیچی

بازی سنگ، کاغذ و قیچی دو بازیکن دارد. بازیکن اول می‌تواند قیچی (کاراکتر 'S')، سنگ (کاراکتر 'R') و یا کاغذ (کاراکتر 'P') را انتخاب کند و بازیکن دوم رایانه است که می‌تواند یکی از کاراکترها را به صورت تصادفی انتخاب نماید. اگر انتخاب دو بازیکن یکسان باشد، مساوی‌اند. اگر متفاوت باشد، یک نفر برنده است و برنده یک امتیاز را کسب می‌کند. برای مثال:

اگر بازیکن اول قیچی و بازیکن دوم کاغذ را انتخاب کند، بازیکن اول برنده است. چون قیچی کاغذ را می‌برد. اگر بازیکن اول قیچی و بازیکن دوم سنگ را انتخاب کند، بازیکن دوم برنده است. زیرا سنگ قیچی را می‌شکند.

اگر بازیکن اول سنگ و بازیکن دوم کاغذ را انتخاب کند، بازیکن دوم برنده است. زیرا کاغذ دور سنگ می پیچد.

برنامه ای که این مسابقه را بین رایانه و یک بازیکن انجام می دهد و برنده را نمایش می دهد.

۱. یک فایل جدید پایتون ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```

1 import random
2 import os
3 import re
4 os.system('cls' if os.name=='nt' else 'clear')
5 while (True):
6     print ("\n")
7     print("Rock, Paper, Scissors - Shoot!")
8     userChoice = input("Choose your weapon [R]ock], [P]aper, or
9         [S]cissors or [E]nd : ")
10    if userChoice in ['e', 'E']: break
11    if not re.match("[SsRrPpeE]", userChoice):
12        print("Please choose a letter:")
13        print ("[R]ock, [S]cissors or [P]aper or [E]nd.")
14        continue
15    # Echo the user's choice
16    print ("You chose: " + userChoice)
17    choices = ['R', 'P', 'S']
18    opponenetChoice = random.choice(choices)
19    print("I chose: " + opponenetChoice)
20    if opponenetChoice == str.upper(userChoice):
21        print ("Tie! ")
22    elif opponenetChoice == 'R' and userChoice.upper() == 'S':
23        print ("Scissors beats rock, I win! ")
24        continue
25    elif opponenetChoice == 'S' and userChoice.upper() == 'P':
26        print ("Scissors beats paper! I win! ")
27        continue
28    elif opponenetChoice == 'P' and userChoice.upper() == 'R':
29        print ("Paper beat rock, I win! ")
30        continue
31    else:
32        print ("You win!")

```

دستورات ۱ تا ۳، بسته های مورد نیاز را به برنامه اضافه می کنند، دستور ۴، صفحه خروجی را پاک می کند، دستور

۵، یک حلقه بی نهایت ایجاد می کند که در این حلقه اعمال زیر انجام می شود:

۱. ابتدا مکان نمای چاپ به سطر بعد می رود (دستور ۶)، سپس یک پیغام نمایش داده می شود (دستور ۷).

۲. دستورات ۸ و ۹، از کاربر یک کاراکتر را دریافت می کنند (یکی از کاراکترهای 'R'، 'P'، 'S' یا 'E').

## پروژه‌های برنامه‌نویسی ساده و کوچک ۱۱

۳. اگر کاراکتر دریافتی 'E' یا 'e' باشد، با دستور break حلقه را خاتمه می‌دهد (دستور ۱۰).
  ۴. اگر کاراکتر دریافتی، یکی از کاراکترهای 'S'، 's'، 'R'، 'r'، 'P'، 'p'، 'E' یا 'e' نباشد، یک پیغام نشان می‌دهد و با دستور continue به ابتدای حلقه برمی‌گردد (دستورات ۱۲ تا ۱۴).
  ۵. کاراکتر وارد شده توسط کاربر را با یک پیغام نمایش می‌دهد (دستور ۱۶).
  ۶. لیست choices را با اعضای ['R', 'P', 'S'] تعریف می‌کند.
  ۷. با متد choice() کلاس random یکی از اعضای لیست choices را انتخاب می‌کند که انتخاب رایانه می‌باشد (دستور ۱۸).
  ۸. با پیغام مناسب نشان می‌دهد که رایانه چه کاراکتری را انتخاب کرده است (دستور ۱۹).
  ۹. اگر هر دو انتخاب برابر باشند، پیغام "Tie!" را نشان می‌دهد (دستورات ۲۰ و ۲۱).
  ۱۰. وگرنه، اگر انتخاب کامپیوتر 'R' و انتخاب کاربر 'S' باشد، پیغام "scissors" را نمایش می‌دهد و به ابتدای حلقه برمی‌گردد (دستورات ۲۲ تا ۲۴).
  ۱۱. وگرنه اگر انتخاب کامپیوتر 'S' و انتخاب کاربر 'P' باشد، پیغام "Scissors beats paper! I Win!" را نمایش می‌دهد و به ابتدای حلقه برمی‌گردد (دستورات ۲۵ تا ۲۷).
  ۱۲. وگرنه، اگر کامپیوتر 'P' را انتخاب کرده باشد و کاربر 'R' را انتخاب کرده باشد، پیغام "Paper beat rock, I Win!" را نمایش می‌دهد و به ابتدای حلقه برمی‌گردد (دستورات ۲۸ تا ۳۰).
  ۱۳. وگرنه پیغام "You Win!" را نمایش می‌دهد (دستورات ۳۱ و ۳۲).
۲. پروژه را اجرا کرده، ورودی‌های خواسته شده را وارد کنید (مانند زیر) و خروجی را مشاهده نمایید:

```
Rock, Paper, Scissors - Shoot!
```

```
Choose your weapon [R]ock], [P]aper, or [S]cissors or [E]nd : p
```

```
You chose: p
```

```
I chose: R
```

```
You win!
```

```
Rock, Paper, Scissors - Shoot!
```

```
Choose your weapon [R]ock], [P]aper, or [S]cissors or [E]nd : s
```

```
You chose: s
```

```
I chose: R
```

```
Scissors beats rock, I win!
```

```
Rock, Paper, Scissors - Shoot!
```

```
Choose your weapon [R]ock], [P]aper, or [S]cissors or [E]nd : s
```

```
You chose: s
```

```
I chose: S
```

```
Tie!
```

Rock, Paper, Scissors - Shoot!  
Choose your weapon [R]ock], [P]aper, or [S]cissors or [E]nd : e

### ۳-۱. پیاده‌سازی برنامه ساخت ماشین حساب ساده

برنامه‌ای که ماشین حساب ساده‌ای ایجاد می‌کند که چهار عمل اصلی محاسباتی جمع، تفریق، ضرب و تقسیم را انجام می‌دهد.

۱. یک فایل جدید پایتون ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```

1 def addition ():
2     print("Addition")
3     n = float(input("Enter the number: "))
4     t = 0 #Total number enter
5     ans = 0
6     while n != 0:
7         ans = ans + n
8         t+=1
9         n = float(input("Enter another number (0 to calculate): "))
10    return [ans,t]
11 def subtraction ():
12    print("Subtraction");
13    n = float(input("Enter the number: "))
14    t = 0 #Total number enter
15    sum = 0
16    ans = n
17    while n != 0:
18        n = float(input("Enter another number (0 to calculate): "))
19        ans = ans - n
20        t+=1
21    return [ans,t]
22 def multiplication ():
23    print("Multiplication")
24    n = float(input("Enter the number: "))
25    t = 0 #Total number enter
26    ans = 1
27    while n != 0:
28        ans = ans * n
29        t+=1
30        n = float(input("Enter another number (0 to calculate): "))
31    return [ans,t]
32 def division():
33    an = []
34    an = addition()

```

پروژه‌های برنامه‌نویسی ساده و کوچک ۱۳

```
35     t = an[1]
36     a = an[0]
37     ans = a / t
38     return [ans,t]
39
40 while True:
41     list = []
42     print()
43     print(" Enter '1', 'a', 'A', '+', 'add', 'Add' for addition")
44     print(" Enter '2', 's', 'S', '-', 'sub', 'Sub' for subtraction")
45     print(" Enter '3', 'm', 'M', '*', 'mul', 'Mul' for multiplication")
46     print(" Enter '4', 'd', 'D', '/', 'div', 'Div' for average")
47     print(" Enter '5', 'q', 'Q', 'quit', 'Quit' for quit")
48     op = input()
49     if op not in ['5', 'q', 'Q', 'quit', 'Quit']:
50         if op in ['1', 'a', 'A', '+', 'add', 'Add']:
51             list = addition()
52             print("Ans = ", list[0], " total inputs ",list[1])
53         elif op in ['2', 's', 'S', '-', 'sub', 'Sub']:
54             list = subtraction()
55             print("Ans = ", list[0], " total inputs ",list[1])
56         elif op in ['3', 'm', 'M', '*', 'mul', 'Mul']:
57             list = multiplication()
58             print("Ans = ", list[0], " total inputs ",list[1])
59         elif op in ['4', 'd', 'D', '/', 'div', 'Div']:
60             list = division()
61             print("Ans = ", list[0], " total inputs ",list[1])
62         else:
63             print ("invilid input")
64     else:
65         break
```

دستورات ۱ تا ۱۰، تابع addition() را تعریف می‌کنند که چند عدد را به‌عنوان ورودی دریافت کرده، آن‌ها را باهم جمع می‌کند. در پایان، تعداد اعداد (t) و حاصل جمع (ans) را برمی‌گرداند، دستورات ۱۱ تا ۲۱، تابع subtraction() را پیاده‌سازی می‌کنند که چند عدد را خوانده، حاصل تفریق آن‌ها و تعداد اعداد وارد شده را برمی‌گرداند، دستورات ۲۲ تا ۳۱، تابع multiplication() را پیاده‌سازی می‌کند که تعدادی عدد را خوانده، حاصل ضرب آن‌ها را محاسبه می‌کند و تعداد اعداد و حاصل ضرب آن‌ها را برمی‌گرداند، دستورات ۳۲ تا ۳۸، تابع division() را پیاده‌سازی می‌کند که تعدادی عدد را خوانده، حاصل تقسیم آن‌ها و تعداد عدد وارد شده را برمی‌گرداند، دستورات ۴۰ تا ۶۵، یک حلقه بی‌نهایت را ایجاد می‌کنند. در این حلقه، اعمال زیر انجام می‌شود:

۱. لیست List را به‌صورت خالی (تهی) تعریف می‌کند (دستور ۴۱).

۲. پیغام مناسبی را نمایش می‌دهد تا کاربر یک عمل را انتخاب کند (دستورات ۴۲ تا ۴۷).  
 ۳. از کاربر می‌خواهد تا عمل موردنظر را وارد کند و در op قرار می‌دهد (دستور ۴۸).  
 ۴. اگر عمل واردشده یکی از مقادیر '5'، 'q'، 'Q'، 'quit' یا 'Quit' نباشد، شرایط زیر را بررسی می‌کند:

❖ اگر عمل واردشده '1'، 'a'، 'A'، 'add' یا 'Add' باشد، نتیجه فراخوانی تابع addition() را در list قرار می‌دهد و با پیغام مناسب عنصر اول (نتیجه جمع) و عنصر دوم لیست (تعداد اعداد جمع شده) را نمایش می‌دهد.

❖ ولی اگر عمل واردشده '2'، 's'، 'S'، 'sub' یا 'Sub' باشد، نتیجه فراخوانی تابع subtraction() را در لیست قرار می‌دهد و عناصر لیست را با پیغام مناسب نمایش می‌دهد.

❖ وگرنه، اگر عمل واردشده '3'، 'm'، 'M'، 'mul' یا 'Mul' باشد، نتیجه فراخوانی تابع multiplication() را در List قرار می‌دهد و مقادیر عناصر List را با پیغام مناسب نمایش می‌دهد.

❖ وگرنه، اگر عمل واردشده '4'، 'd'، 'D'، 'div' یا 'Div' باشد، نتیجه فراخوانی divition() را در List قرار می‌دهد و مقادیر عناصر List را با پیغام مناسب نمایش می‌دهد.

❖ وگرنه، پیغام "invalid input" را نمایش می‌دهد.

۵. وگرنه با دستور break حلقه را خاتمه می‌دهد.

۲. پروژه را اجرا کرده، ورودی‌های خواسته‌شده را وارد کنید (مانند زیر) و خروجی را مشاهده نمایید:

```

Enter '1', 'a', 'A', '+', 'add', 'Add' for addition
Enter '2', 's', 'S', '-', 'sub', 'Sub' for subtraction
Enter '3', 'm', 'M', '*', 'mul', 'Mul' for multiplication
Enter '4', 'd', 'D', '/', 'div', 'Div' for average
Enter '5', 'q', 'Q', 'quit', 'Quit' for quit
sub
Subtraction
Enter the number: 10
Enter another number (0 to calculate): 11
Enter another number (0 to calculate): 7
Enter another number (0 to calculate): 9
Enter another number (0 to calculate): 0
Ans = -17.0 total inputs 4

Enter '1', 'a', 'A', '+', 'add', 'Add' for addition
Enter '2', 's', 'S', '-', 'sub', 'Sub' for subtraction
Enter '3', 'm', 'M', '*', 'mul', 'Mul' for multiplication
Enter '4', 'd', 'D', '/', 'div', 'Div' for average
Enter '5', 'q', 'Q', 'quit', 'Quit' for quit
    
```

پروژه‌های برنامه‌نویسی ساده و کوچک ۱۵

```
3
Multiplication
Enter the number: 12
Enter another number (0 to calculate): 8
Enter another number (0 to calculate): 2
Enter another number (0 to calculate): 0
Ans = 192.0 total inputs 3

Enter '1', 'a', 'A', '+', 'add', 'Add' for addition
Enter '2', 's', 'S', '-', 'sub', 'Sub' for subtraction
Enter '3', 'm', 'M', '*', 'mul', 'Mul' for multiplication
Enter '4', 'd', 'D', '/', 'div', 'Div' for average
Enter '5', 'q', 'Q', 'quit', 'Quit' for quit
quit
```

## ۴-۱. پیاده‌سازی برنامه جمع و تفریق اعداد بزرگ

برنامه‌ای که دو عدد حداقل ۳۰ رقمی را با هم جمع یا از هم تفریق می‌نماید و نتیجه را نشان می‌دهد.

مراحل طراحی و اجرا

۱. یک فایل جدید پایتون ایجاد کرده، دستورات آن را به صورت زیر تغییر دهید:

```
1 def add(number1, number2):
2     carry = 0
3     result = ""
4     for i in range(len(number1) - 1, -1, -1):
5         r = int(number1[i]) + int(number2[i]) + carry
6         if r >= 10:
7             r = r - 10
8             carry = 1
9         else:
10            carry = 0
11            result = str(r) + result
12    if carry == 1:
13        result = "1" + result
14    return result
15
16 def sub(number1, number2):
17     borrow = 0
18     swap = False
19     if number1[0] < number2[0]:
20         number1, number2 = number2, number1
21         swap = True
22     elif number1[0] == number2[0]:
23         for i in range(1, len(number1)):
24             if number1[i] > number2[i]:
```

```

25         break;
26         elif number1[i] < number2[i]:
27             number1, number2 = number2, number1
28             swap = True
29     result = ""
30     for i in range(len(number1) - 1, -1, -1):
31         r = int(number1[i]) - int(number2[i]) + borrow
32         if r < 0:
33             if i != 0:
34                 r = r + 10
35                 borrow = -1
36             else:
37                 borrow = 0
38         result = str(r) + result
39     if swap == True:
40         result = "-" + result
41     return result
42
43 number1 = input("Enter number1:")
44 number2 = input("Enter number2:")
45 if number1.isdecimal() == False or number2.isdigit() == False:
46     print("Invalid input")
47 else:
48     len1 = len(number1)
49     len2 = len(number2)
50     if len1 < len2:
51         number1 = number1.zfill(len2)
52     else:
53         number2 = number2.zfill(len1)
54     result = add(number1, number2)
55     len3 = len(result)
56     print("\n")
57     number11 = number1.zfill(len3)
58     number21 = number2.zfill(len3)
59     print(number11, " + ")
60     print(number21)
61     print('='*len3)
62     print(result)
63     result = sub(number1, number2)
64     len3 = len(result)
65     print("\n")
66     number11 = number1.zfill(len3)
67     number21 = number2.zfill(len3)
68     print(number11, " - ")
69     print(number21)
70     print('='*len3)

```

71	<b>print(result)</b>
----	----------------------

دستورات ۱ تا ۱۴، تابع `add()` را پیاده‌سازی می‌کنند که دو عدد `number1` و `number2` را به‌عنوان پارامتر دریافت کرده، حاصل جمع آن‌ها را برمی‌گرداند. برای این منظور، اعمال زیر را انجام می‌دهد:

۱. `carry` (رقم نقلی) را برابر ۰ قرار می‌دهد (دستور ۲).
۲. نتیجه جمع `result` را خالی می‌کند (دستور ۳).
۳. ارقام را از سمت راست به چپ پردازش می‌کند (دستور ۴) و اعمال زیر را انجام می‌دهد:
  - ❖ رقم فعلی `number1` و `number2` به‌علاوه `carry` را باهم جمع کرده، نتیجه را در `r` (جمع رقم فعلی) قرار می‌دهد (دستور ۵).
  - ❖ اگر `r` بزرگ‌تر یا مساوی از ۱۰ باشد، ۱۰ واحد از `r` کم کرده، رقم نقلی (`carry`) را برابر ۱ قرار می‌دهد و `carry` را برابر ۰ قرار می‌دهد (دستورات ۶ تا ۱۰).
  - ❖ رقم فعلی به‌دست آمده را به ابتدای `result` اضافه می‌کند (دستور ۱۱).
۱. اگر `carry` برابر ۱ باشد، "۱" را به ابتدای `result` اضافه می‌کند (دستورات ۱۲ و ۱۳).
۲. مقدار `result` را برمی‌گرداند (دستور ۱۴).

دستورات ۱۶ تا ۴۱، تابع `sub()` را تعریف کرده که دو عدد `number1` و `number2` را به‌عنوان پارامتر دریافت کرده، حاصل تفریق آن‌ها را برمی‌گرداند. برای این منظور، اعمال زیر را انجام می‌دهد:

۱. `brrow` (رقم قرضی) را برابر ۰ قرار می‌دهد (دستور ۱۷).
۲. `swap` را برابر `False` قرار می‌دهد (این متغیر تعیین می‌کند که عدد اول کوچک‌تر از عدد دوم بوده و جای‌شان عوض شده است یا نه) (دستور ۱۸).
۳. دستورات ۱۹ تا ۲۸، اگر عدد اول کوچک‌تر از عدد دوم باشد، جای‌شان را عوض کرده و `swap` را برابر `True` قرار می‌دهد.
۴. دستورات ۲۹ تا ۳۸، دو عدد را از هم کم کرده (ارقام را نظیر به نظیر از هم کم می‌نماید) و در `result` قرار می‌دهد.
۵. دستورات ۳۹ و ۴۰، اگر `swap` برابر `True` باشد (یعنی، عدد اول کوچک‌تر از عدد دوم باشد)، یک منفی (-) در قیل از `result` قرار می‌دمد.
۶. مقدار `result` را برمی‌گرداند (دستور ۴۱).

دستور ۴۳ و ۴۴، دو عدد را به‌صورت رشته‌ای می‌خواند، دستورات ۴۵ و ۴۶، اگر در اعداد وارد شده غیر از ارقام ۰

تا ۹ باشد، پیغام "Invalid input" را نمایش می‌دهد، و گرنه اعمال زیر را انجام می‌دهد:

۱. دستورات ۴۸ و ۴۹، طول اعداد وارد شده `number1` و `number2` را به ترتیب در `len1` و `len2` قرار می‌دهند.

۲. دستورات ۵۰ و ۵۱، طول دو عدد number1 و number2 را یکی می‌کنند. یعنی، هر عددی که طول آن کم تر بود، یا تابع zfill() به ابتدای آن "0" اضافه می‌کند.

۳. دستورات ۵۴ تا ۶۲، ابتدا با فراخوانی تابع add()، دو عدد را جمع کرده، نتیجه را در result قرار می‌دهند، سپس، طول result را حساب کرده، اعداد number1 و number2 را به طول result تغییر می‌دهند و در number11 و number12 قرار می‌دهند و در نهایت، number11، number12 و result را نمایش می‌دهند.

دستورات ۶۳ تا ۷۱، ابتدا با فراخوانی تابع sub()، دو عدد را از هم کم کرده، نتیجه را در result قرار می‌دهند، سپس، طول result را حساب کرده، اعداد number1، number2 را به طول result تغییر داده، در number11 و number12 قرار می‌دهند و در نهایت، number11، number12 و result را نمایش می‌دهند.

۲. پروژه را اجرا کرده، دو عدد را وارد کنید(مانند زیر) و خروجی را مشاهده نمایید:

```
Enter number1:8888888888888878272262662626626
Enter number2:3888888888888902992929292929229

00888888888888888878272262662626626626   +
38888888888889029929292929292929229
=====
397777777777790820155565555555855

00088888888888888878272262662626626626   -
03888888888889029929292929292929229
=====
-3800000000000015165703033030302603
```

## ۵-۱. پیاده‌سازی برنامه دریافت روزی از سال، نمایش ماه و فصل

برنامه‌ای که روزی از سال را خوانده، مشخص می‌کند که چه فصلی، چه ماه و چه روزی از سال است. به‌عنوان مثال، روز ۳۵ سال، فصل بهار، ماه اردیبهشت و روز چهارم این ماه است.

```
1 day= int(input("Enter number of a day: "))
2 if (day < 1 or day > 365):
3     print ("Day should be between 1 and 365)")
4 else:
5     month = day // 31
6     if month > 6:
7         month = 6
8     day -= month * 31
9     if day != 0:
```

پروژه‌های برنامه‌نویسی ساده و کوچک ۱۹

```
10     count = day // 30
11     month += count + 1
12     day -= count * 30
13     else:
14         day = 31
15     if day == 0:
16         day = 30
17         month = month - 1
18     print("day ", day, " of ", month, " month ", end='')
19     if month == 1:
20         print("(Farvardin) ", end='')
21     elif month == 2:
22         print("(Ordibehesht) ", end='')
23     elif month == 3:
24         print("(Khordad) ", end='')
25     elif month == 4:
26         print("(Tir) ", end='')
27     elif month == 5:
28         print("(Mordad) ", end='')
29     elif month == 6:
30         print("(Shahrivar) ", end='')
31     elif month == 7:
32         print("(Mehr) ", end='')
33     elif month == 8:
34         print("(Aban) ", end='')
35     elif month == 9:
36         print("(Azar) ", end='')
37     elif month == 10:
38         print("(Day) ", end='')
39     elif month == 11:
40         print("(Bahman) ", end='')
41     elif month == 12:
42         print("(Esfand) ", end='')
43     print(" of season ", end='')
44     if month > 9:
45         print("Winter ", end='')
46     elif month > 6 :
47         print("Autumn ", end='')
48     elif month > 3:
49         print("Summer ", end='')
50     else:
51         print("Spring ", end='')
```

Enter number of a day: 321

day 15 of 11 month (Bahman) of season Winter

## ۶-۱. پیاده‌سازی برنامه محاسبه مقدار یک سری پیچیده با حلقه تکرار

برنامه‌ای که  $x$  و  $n$  را خوانده، حاصل سری زیر را محاسبه می‌کند.

$$\frac{x^n n^x (x+n)^{n+x}}{n! x! (n+x)!} \sum_{i=1}^n \frac{x^i i^x}{n! i! (n+i)!}$$

```

1 x = int(input("Enter x:"))
2 n = int(input("Enter n:"))
3 p1 = 1
4 for i in range(1, n + 1):
5     p1 = p1 * x
6 p2 = 1
7 for i in range(1, x + 1):
8     p2 = p2 * n
9 p3 = 1
10 for i in range(1, n + x + 1):
11     p3 = p3 * (x + n)
12 f1 = 1
13 for i in range(1, n + 1):
14     f1 = f1 * I
15 f2 = 1
16 for i in range(1, x + 1):
17     f2 = f2 * i
18 f3 = 1
19 for i in range(1, n + x + 1):
20     f3 = f3 * i
21 result1 = (p1 * p2 * p3) / (f1 * f2 * f3)
22 sum = 0
23 for i in range(1, n + 1):
24     p4 = 1
25     for j in range(1, i + 1):
26         p4 = p4 * x
27     p5 = 1
28     for j in range(1, x+1):
29         p5 = p5 * j
30     f4 = 1
31     for j in range(1, n + 1):
32         f4 = f4 * j
33     f5 = 1
34     for j in range(1, i + 1):
35         f5 = f5 * j
36     f6 = 1
37     for j in range(1, n + i + 1):
38         f6 = f6 * j
39     sum = sum + (p4 * p5) / (f4 * f5 * f6)
40 result = result1 * sum

```

پروژه‌های برنامه‌نویسی ساده و کوچک ۲۱

```
41 print("Result is ", result)
Enter x:12
Enter n:7
Result is 191200853982517.25
```

## ۷-۱. پیاده‌سازی برنامه نمایش جفت اعداد متحابه

برنامه‌ای که تمامی جفت‌های اعداد متحابه از یک تا عدد خوانده شده را نمایش می‌دهد. دو عدد صحیح متحابه گویند، اگر مجموع مقسوم‌علیه‌های هر یک به جز خود عدد با عدد دوم برابر باشد. به‌عنوان مثال، مقسوم‌علیه‌های ۲۲۰ که از ۲۲۰ کوچک‌تر هستند، عبارت‌اند از: ۱، ۲، ۳، ۴، ۵، ۶، ۱۰، ۱۱، ۱۲، ۱۳، ۱۴، ۱۵، ۱۶، ۲۰، ۲۲، ۲۳، ۲۴، ۲۵، ۲۶، ۳۰، ۳۳، ۳۴، ۳۵، ۳۶، ۴۰، ۴۲، ۴۳، ۴۴، ۴۵، ۴۶، ۴۷، ۴۸، ۴۹، ۵۰، ۵۱، ۵۲، ۵۳، ۵۴، ۵۵، ۵۶، ۵۷، ۵۸، ۵۹، ۶۰، ۶۱، ۶۲، ۶۳، ۶۴، ۶۵، ۶۶، ۶۷، ۶۸، ۶۹، ۷۰، ۷۱، ۷۲، ۷۳، ۷۴، ۷۵، ۷۶، ۷۷، ۷۸، ۷۹، ۸۰، ۸۱، ۸۲، ۸۳، ۸۴، ۸۵، ۸۶، ۸۷، ۸۸، ۸۹، ۹۰، ۹۱، ۹۲، ۹۳، ۹۴، ۹۵، ۹۶، ۹۷، ۹۸، ۹۹، ۱۰۰، ۱۰۱، ۱۰۲، ۱۰۳، ۱۰۴، ۱۰۵، ۱۰۶، ۱۰۷، ۱۰۸، ۱۰۹، ۱۱۰، ۱۱۱، ۱۱۲، ۱۱۳، ۱۱۴، ۱۱۵، ۱۱۶، ۱۱۷، ۱۱۸، ۱۱۹، ۱۲۰، ۱۲۱، ۱۲۲، ۱۲۳، ۱۲۴، ۱۲۵، ۱۲۶، ۱۲۷، ۱۲۸، ۱۲۹، ۱۳۰، ۱۳۱، ۱۳۲، ۱۳۳، ۱۳۴، ۱۳۵، ۱۳۶، ۱۳۷، ۱۳۸، ۱۳۹، ۱۴۰، ۱۴۱، ۱۴۲، ۱۴۳، ۱۴۴، ۱۴۵، ۱۴۶، ۱۴۷، ۱۴۸، ۱۴۹، ۱۵۰، ۱۵۱، ۱۵۲، ۱۵۳، ۱۵۴، ۱۵۵، ۱۵۶، ۱۵۷، ۱۵۸، ۱۵۹، ۱۶۰، ۱۶۱، ۱۶۲، ۱۶۳، ۱۶۴، ۱۶۵، ۱۶۶، ۱۶۷، ۱۶۸، ۱۶۹، ۱۷۰، ۱۷۱، ۱۷۲، ۱۷۳، ۱۷۴، ۱۷۵، ۱۷۶، ۱۷۷، ۱۷۸، ۱۷۹، ۱۸۰، ۱۸۱، ۱۸۲، ۱۸۳، ۱۸۴، ۱۸۵، ۱۸۶، ۱۸۷، ۱۸۸، ۱۸۹، ۱۹۰، ۱۹۱، ۱۹۲، ۱۹۳، ۱۹۴، ۱۹۵، ۱۹۶، ۱۹۷، ۱۹۸، ۱۹۹، ۲۰۰، ۲۰۱، ۲۰۲، ۲۰۳، ۲۰۴، ۲۰۵، ۲۰۶، ۲۰۷، ۲۰۸، ۲۰۹، ۲۱۰، ۲۱۱، ۲۱۲، ۲۱۳، ۲۱۴، ۲۱۵، ۲۱۶، ۲۱۷، ۲۱۸، ۲۱۹، ۲۲۰، ۲۲۱، ۲۲۲، ۲۲۳، ۲۲۴، ۲۲۵، ۲۲۶، ۲۲۷، ۲۲۸، ۲۲۹، ۲۳۰، ۲۳۱، ۲۳۲، ۲۳۳، ۲۳۴، ۲۳۵، ۲۳۶، ۲۳۷، ۲۳۸، ۲۳۹، ۲۴۰، ۲۴۱، ۲۴۲، ۲۴۳، ۲۴۴، ۲۴۵، ۲۴۶، ۲۴۷، ۲۴۸، ۲۴۹، ۲۵۰، ۲۵۱، ۲۵۲، ۲۵۳، ۲۵۴، ۲۵۵، ۲۵۶، ۲۵۷، ۲۵۸، ۲۵۹، ۲۶۰، ۲۶۱، ۲۶۲، ۲۶۳، ۲۶۴، ۲۶۵، ۲۶۶، ۲۶۷، ۲۶۸، ۲۶۹، ۲۷۰، ۲۷۱، ۲۷۲، ۲۷۳، ۲۷۴، ۲۷۵، ۲۷۶، ۲۷۷، ۲۷۸، ۲۷۹، ۲۸۰، ۲۸۱، ۲۸۲، ۲۸۳، ۲۸۴، ۲۸۵، ۲۸۶، ۲۸۷، ۲۸۸، ۲۸۹، ۲۹۰، ۲۹۱، ۲۹۲، ۲۹۳، ۲۹۴، ۲۹۵، ۲۹۶، ۲۹۷، ۲۹۸، ۲۹۹، ۳۰۰، ۳۰۱، ۳۰۲، ۳۰۳، ۳۰۴، ۳۰۵، ۳۰۶، ۳۰۷، ۳۰۸، ۳۰۹، ۳۱۰، ۳۱۱، ۳۱۲، ۳۱۳، ۳۱۴، ۳۱۵، ۳۱۶، ۳۱۷، ۳۱۸، ۳۱۹، ۳۲۰، ۳۲۱، ۳۲۲، ۳۲۳، ۳۲۴، ۳۲۵، ۳۲۶، ۳۲۷، ۳۲۸، ۳۲۹، ۳۳۰، ۳۳۱، ۳۳۲، ۳۳۳، ۳۳۴، ۳۳۵، ۳۳۶، ۳۳۷، ۳۳۸، ۳۳۹، ۳۴۰، ۳۴۱، ۳۴۲، ۳۴۳، ۳۴۴، ۳۴۵، ۳۴۶، ۳۴۷، ۳۴۸، ۳۴۹، ۳۵۰، ۳۵۱، ۳۵۲، ۳۵۳، ۳۵۴، ۳۵۵، ۳۵۶، ۳۵۷، ۳۵۸، ۳۵۹، ۳۶۰، ۳۶۱، ۳۶۲، ۳۶۳، ۳۶۴، ۳۶۵، ۳۶۶، ۳۶۷، ۳۶۸، ۳۶۹، ۳۷۰، ۳۷۱، ۳۷۲، ۳۷۳، ۳۷۴، ۳۷۵، ۳۷۶، ۳۷۷، ۳۷۸، ۳۷۹، ۳۸۰، ۳۸۱، ۳۸۲، ۳۸۳، ۳۸۴، ۳۸۵، ۳۸۶، ۳۸۷، ۳۸۸، ۳۸۹، ۳۹۰، ۳۹۱، ۳۹۲، ۳۹۳، ۳۹۴، ۳۹۵، ۳۹۶، ۳۹۷، ۳۹۸، ۳۹۹، ۴۰۰، ۴۰۱، ۴۰۲، ۴۰۳، ۴۰۴، ۴۰۵، ۴۰۶، ۴۰۷، ۴۰۸، ۴۰۹، ۴۱۰، ۴۱۱، ۴۱۲، ۴۱۳، ۴۱۴، ۴۱۵، ۴۱۶، ۴۱۷، ۴۱۸، ۴۱۹، ۴۲۰، ۴۲۱، ۴۲۲، ۴۲۳، ۴۲۴، ۴۲۵، ۴۲۶، ۴۲۷، ۴۲۸، ۴۲۹، ۴۳۰، ۴۳۱، ۴۳۲، ۴۳۳، ۴۳۴، ۴۳۵، ۴۳۶، ۴۳۷، ۴۳۸، ۴۳۹، ۴۴۰، ۴۴۱، ۴۴۲، ۴۴۳، ۴۴۴، ۴۴۵، ۴۴۶، ۴۴۷، ۴۴۸، ۴۴۹، ۴۵۰، ۴۵۱، ۴۵۲، ۴۵۳، ۴۵۴، ۴۵۵، ۴۵۶، ۴۵۷، ۴۵۸، ۴۵۹، ۴۶۰، ۴۶۱، ۴۶۲، ۴۶۳، ۴۶۴، ۴۶۵، ۴۶۶، ۴۶۷، ۴۶۸، ۴۶۹، ۴۷۰، ۴۷۱، ۴۷۲، ۴۷۳، ۴۷۴، ۴۷۵، ۴۷۶، ۴۷۷، ۴۷۸، ۴۷۹، ۴۸۰، ۴۸۱، ۴۸۲، ۴۸۳، ۴۸۴، ۴۸۵، ۴۸۶، ۴۸۷، ۴۸۸، ۴۸۹، ۴۹۰، ۴۹۱، ۴۹۲، ۴۹۳، ۴۹۴، ۴۹۵، ۴۹۶، ۴۹۷، ۴۹۸، ۴۹۹، ۵۰۰، ۵۰۱، ۵۰۲، ۵۰۳، ۵۰۴، ۵۰۵، ۵۰۶، ۵۰۷، ۵۰۸، ۵۰۹، ۵۱۰، ۵۱۱، ۵۱۲، ۵۱۳، ۵۱۴، ۵۱۵، ۵۱۶، ۵۱۷، ۵۱۸، ۵۱۹، ۵۲۰، ۵۲۱، ۵۲۲، ۵۲۳، ۵۲۴، ۵۲۵، ۵۲۶، ۵۲۷، ۵۲۸، ۵۲۹، ۵۳۰، ۵۳۱، ۵۳۲، ۵۳۳، ۵۳۴، ۵۳۵، ۵۳۶، ۵۳۷، ۵۳۸، ۵۳۹، ۵۴۰، ۵۴۱، ۵۴۲، ۵۴۳، ۵۴۴، ۵۴۵، ۵۴۶، ۵۴۷، ۵۴۸، ۵۴۹، ۵۵۰، ۵۵۱، ۵۵۲، ۵۵۳، ۵۵۴، ۵۵۵، ۵۵۶، ۵۵۷، ۵۵۸، ۵۵۹، ۵۶۰، ۵۶۱، ۵۶۲، ۵۶۳، ۵۶۴، ۵۶۵، ۵۶۶، ۵۶۷، ۵۶۸، ۵۶۹، ۵۷۰، ۵۷۱، ۵۷۲، ۵۷۳، ۵۷۴، ۵۷۵، ۵۷۶، ۵۷۷، ۵۷۸، ۵۷۹، ۵۸۰، ۵۸۱، ۵۸۲، ۵۸۳، ۵۸۴، ۵۸۵، ۵۸۶، ۵۸۷، ۵۸۸، ۵۸۹، ۵۹۰، ۵۹۱، ۵۹۲، ۵۹۳، ۵۹۴، ۵۹۵، ۵۹۶، ۵۹۷، ۵۹۸، ۵۹۹، ۶۰۰، ۶۰۱، ۶۰۲، ۶۰۳، ۶۰۴، ۶۰۵، ۶۰۶، ۶۰۷، ۶۰۸، ۶۰۹، ۶۱۰، ۶۱۱، ۶۱۲، ۶۱۳، ۶۱۴، ۶۱۵، ۶۱۶، ۶۱۷، ۶۱۸، ۶۱۹، ۶۲۰، ۶۲۱، ۶۲۲، ۶۲۳، ۶۲۴، ۶۲۵، ۶۲۶، ۶۲۷، ۶۲۸، ۶۲۹، ۶۳۰، ۶۳۱، ۶۳۲، ۶۳۳، ۶۳۴، ۶۳۵، ۶۳۶، ۶۳۷، ۶۳۸، ۶۳۹، ۶۴۰، ۶۴۱، ۶۴۲، ۶۴۳، ۶۴۴، ۶۴۵، ۶۴۶، ۶۴۷، ۶۴۸، ۶۴۹، ۶۵۰، ۶۵۱، ۶۵۲، ۶۵۳، ۶۵۴، ۶۵۵، ۶۵۶، ۶۵۷، ۶۵۸، ۶۵۹، ۶۶۰، ۶۶۱، ۶۶۲، ۶۶۳، ۶۶۴، ۶۶۵، ۶۶۶، ۶۶۷، ۶۶۸، ۶۶۹، ۶۷۰، ۶۷۱، ۶۷۲، ۶۷۳، ۶۷۴، ۶۷۵، ۶۷۶، ۶۷۷، ۶۷۸، ۶۷۹، ۶۸۰، ۶۸۱، ۶۸۲، ۶۸۳، ۶۸۴، ۶۸۵، ۶۸۶، ۶۸۷، ۶۸۸، ۶۸۹، ۶۹۰، ۶۹۱، ۶۹۲، ۶۹۳، ۶۹۴، ۶۹۵، ۶۹۶، ۶۹۷، ۶۹۸، ۶۹۹، ۷۰۰، ۷۰۱، ۷۰۲، ۷۰۳، ۷۰۴، ۷۰۵، ۷۰۶، ۷۰۷، ۷۰۸، ۷۰۹، ۷۱۰، ۷۱۱، ۷۱۲، ۷۱۳، ۷۱۴، ۷۱۵، ۷۱۶، ۷۱۷، ۷۱۸، ۷۱۹، ۷۲۰، ۷۲۱، ۷۲۲، ۷۲۳، ۷۲۴، ۷۲۵، ۷۲۶، ۷۲۷، ۷۲۸، ۷۲۹، ۷۳۰، ۷۳۱، ۷۳۲، ۷۳۳، ۷۳۴، ۷۳۵، ۷۳۶، ۷۳۷، ۷۳۸، ۷۳۹، ۷۴۰، ۷۴۱، ۷۴۲، ۷۴۳، ۷۴۴، ۷۴۵، ۷۴۶، ۷۴۷، ۷۴۸، ۷۴۹، ۷۵۰، ۷۵۱، ۷۵۲، ۷۵۳، ۷۵۴، ۷۵۵، ۷۵۶، ۷۵۷، ۷۵۸، ۷۵۹، ۷۶۰، ۷۶۱، ۷۶۲، ۷۶۳، ۷۶۴، ۷۶۵، ۷۶۶، ۷۶۷، ۷۶۸، ۷۶۹، ۷۷۰، ۷۷۱، ۷۷۲، ۷۷۳، ۷۷۴، ۷۷۵، ۷۷۶، ۷۷۷، ۷۷۸، ۷۷۹، ۷۸۰، ۷۸۱، ۷۸۲، ۷۸۳، ۷۸۴، ۷۸۵، ۷۸۶، ۷۸۷، ۷۸۸، ۷۸۹، ۷۹۰، ۷۹۱، ۷۹۲، ۷۹۳، ۷۹۴، ۷۹۵، ۷۹۶، ۷۹۷، ۷۹۸، ۷۹۹، ۸۰۰، ۸۰۱، ۸۰۲، ۸۰۳، ۸۰۴، ۸۰۵، ۸۰۶، ۸۰۷، ۸۰۸، ۸۰۹، ۸۱۰، ۸۱۱، ۸۱۲، ۸۱۳، ۸۱۴، ۸۱۵، ۸۱۶، ۸۱۷، ۸۱۸، ۸۱۹، ۸۲۰، ۸۲۱، ۸۲۲، ۸۲۳، ۸۲۴، ۸۲۵، ۸۲۶، ۸۲۷، ۸۲۸، ۸۲۹، ۸۳۰، ۸۳۱، ۸۳۲، ۸۳۳، ۸۳۴، ۸۳۵، ۸۳۶، ۸۳۷، ۸۳۸، ۸۳۹، ۸۴۰، ۸۴۱، ۸۴۲، ۸۴۳، ۸۴۴، ۸۴۵، ۸۴۶، ۸۴۷، ۸۴۸، ۸۴۹، ۸۵۰، ۸۵۱، ۸۵۲، ۸۵۳، ۸۵۴، ۸۵۵، ۸۵۶، ۸۵۷، ۸۵۸، ۸۵۹، ۸۶۰، ۸۶۱، ۸۶۲، ۸۶۳، ۸۶۴، ۸۶۵، ۸۶۶، ۸۶۷، ۸۶۸، ۸۶۹، ۸۷۰، ۸۷۱، ۸۷۲، ۸۷۳، ۸۷۴، ۸۷۵، ۸۷۶، ۸۷۷، ۸۷۸، ۸۷۹، ۸۸۰، ۸۸۱، ۸۸۲، ۸۸۳، ۸۸۴، ۸۸۵، ۸۸۶، ۸۸۷، ۸۸۸، ۸۸۹، ۸۹۰، ۸۹۱، ۸۹۲، ۸۹۳، ۸۹۴، ۸۹۵، ۸۹۶، ۸۹۷، ۸۹۸، ۸۹۹، ۹۰۰، ۹۰۱، ۹۰۲، ۹۰۳، ۹۰۴، ۹۰۵، ۹۰۶، ۹۰۷، ۹۰۸، ۹۰۹، ۹۱۰، ۹۱۱، ۹۱۲، ۹۱۳، ۹۱۴، ۹۱۵، ۹۱۶، ۹۱۷، ۹۱۸، ۹۱۹، ۹۲۰، ۹۲۱، ۹۲۲، ۹۲۳، ۹۲۴، ۹۲۵، ۹۲۶، ۹۲۷، ۹۲۸، ۹۲۹، ۹۳۰، ۹۳۱، ۹۳۲، ۹۳۳، ۹۳۴، ۹۳۵، ۹۳۶، ۹۳۷، ۹۳۸، ۹۳۹، ۹۴۰، ۹۴۱، ۹۴۲، ۹۴۳، ۹۴۴، ۹۴۵، ۹۴۶، ۹۴۷، ۹۴۸، ۹۴۹، ۹۵۰، ۹۵۱، ۹۵۲، ۹۵۳، ۹۵۴، ۹۵۵، ۹۵۶، ۹۵۷، ۹۵۸، ۹۵۹، ۹۶۰، ۹۶۱، ۹۶۲، ۹۶۳، ۹۶۴، ۹۶۵، ۹۶۶، ۹۶۷، ۹۶۸، ۹۶۹، ۹۷۰، ۹۷۱، ۹۷۲، ۹۷۳، ۹۷۴، ۹۷۵، ۹۷۶، ۹۷۷، ۹۷۸، ۹۷۹، ۹۸۰، ۹۸۱، ۹۸۲، ۹۸۳، ۹۸۴، ۹۸۵، ۹۸۶، ۹۸۷، ۹۸۸، ۹۸۹، ۹۹۰، ۹۹۱، ۹۹۲، ۹۹۳، ۹۹۴، ۹۹۵، ۹۹۶، ۹۹۷، ۹۹۸، ۹۹۹، ۱۰۰۰.

```
1 def sumDiv (n):
2     sum = 0
3     for j in range(1, n // 2 + 1):
4         if n % j == 0:
5             sum += j
6     return sum
7 def main():
8     n = int(input("Enter n:"))
9     for i in range(3, n+1):
10        sum = sumDiv (i)
11        if i == sumDiv (sum) and i != sum:
12            print("Numbers ", i, " and ", sum, " are friendly")
13 main()
```

➤ تابع `main()`، با یک پیغام مناسب `n` را از ورودی دریافت می‌کند. برنامه از ۳ تا `n` را شمارش می‌کند (با استفاده از یک حلقه)، در داخل حلقه مجموع مقسوم‌علیه‌های `n` را با فراخوانی تابع `sumDiv` محاسبه کرده، در `sum` قرار می‌دهد. اکنون، اگر مجموع مقسوم‌علیه‌های `sum` مساوی `i` و `i` مخالف `sum` باشد، آن عدد متحابه است و با پیغام مناسب عدد `i` و `sum` را نمایش می‌دهد.

➤ تابع `sumDiv()`، `n` را به‌عنوان پارامتر دریافت کرده، مجموع مقسوم‌علیه‌های آن را محاسبه می‌کند و برمی‌گرداند.

```
Enter n:1000
Numbers 220 and 284 are friendly
Numbers 284 and 220 are friendly
```

## ۸-۱. پیاده‌سازی برنامه انجام عملیات مختلف بر روی آرایه با توابع

برنامه‌ای که دو آرایه  $3 \times 3$  را خوانده، دو آرایه را با هم جمع، تفریق و ضرب می‌نماید. در این برنامه امکاناتی برای نمایش دو آرایه دارد. در ضمن این برنامه می‌تواند `N`، `M` و `k` را بخواند. سپس حاصل عبارت  $N * A - M * B + K$  را محاسبه کند و نمایش دهد (`A` و `B` به ترتیب آرایه‌های اول و دوم هستند).

```

1 import numpy.matlib
2 import numpy as np
3 def readA2D (a, n, m):
4     for i in range(0, n):
5         print("Enter ", n, "numbers for row ", i + 1, end = " :")
6         s = input().split(' ')
7         for j in range(0, n):
8             a[i, j] = (s[j])
9 def printA2D(a, n, m):
10    for i in range (0, n):
11        for j in range (0, n):
12            print("  ", a[i, j], end = '')
13        print()
14 def addArr (arr1, arr2, arr3, n, m):
15    for i in range(0, n):
16        for j in range(0, m):
17            arr3[i, j] = arr1[i, j] + arr2[i, j]
18 def subArr(arr1, arr2, arr3, n, m):
19    for i in range (0, n):
20        for j in range (0, m):
21            arr3[i, j] = arr1[i, j] - arr2[i, j]
22 def mulArr (arr1, arr2, arr3, n, m):
23    for i in range(0, n):
24        for j in range(0, m):
25            s = 0
26            for k in range(0, m):
27                s = s + arr1[i, k] * arr2[k, j]
28            arr3[i, j] = s
29 def scalarMulArr( arr1, arr2, n, m, value):
30    for i in range(0, n):
31        for j in range(0, m):
32            arr2[i, j] = arr1[i, j] * value
33 def scalarCalArr (arr1, arr2, n, m, value):
34    for i in range(0, n):
35        for j in range(0, m):
36            arr2[i, j] = arr1[i, j] + value
37 def menu():
38    print("1.Read array A")
39    print("2.Read array B")
40    print("3.Display array A")
41    print("4.Display array B")
42    print("5.Add array ")
43    print("6.Sub array ")
44    print("7.Mul array ")
45    print("8. N * A - M * B + K")
46    print("9.Exit ")

```

```
47     choice = int(input("Select 1 to 9: "))
48     return choice
49 def main():
50     n = 3; m = 3
51     arr1 = np.matlib.zeros ([n, m])
52     arr2 = np.matlib.zeros ([n, m])
53     arr3 = np.matlib.zeros ([n, m])
54     while True:
55         choice = menu ()
56         if choice == 1:
57             readA2D (arr1, n, m)
58         elif choice == 2:
59             readA2D (arr2, n, m)
60         elif choice == 3:
61             print(" A is ")
62             printA2D (arr1, n, m)
63         elif choice == 4:
64             print(" B is ")
65             printA2D (arr2, n, m)
66         elif choice == 5:
67             print (" A is ")
68             printA2D (arr1, n, m)
69             print (" B is ")
70             printA2D (arr2, n, m)
71             print (" A + B =")
72             addArr (arr1, arr2, arr3, n, m)
73             printA2D (arr3, n, m)
74         elif choice == 6:
75             print (" A is ")
76             printA2D (arr1, n, m)
77             print (" B is ")
78             printA2D (arr2, n, m)
79             print ("A - B = ")
80             subArr (arr1, arr2, arr3, n, m)
81             printA2D (arr3, n, m)
82         elif choice == 7:
83             print (" A is ")
84             printA2D (arr1, n, m)
85             print (" B is ")
86             printA2D (arr2, n, m)
87             print (" A * B = ")
88             mulArr (arr1, arr2, arr3, n, m)
89             printA2D (arr3, n, m)
90         elif choice == 8:
91             inp = input("Enter N, M, K:").split(' ')
92             N = int(inp[0])
```

```

93     M = int(inp[1])
94     K = int(inp[2])
95     scalarMulArr (arr1, arr3, n, m, N)
96     print(" A is ")
97     printA2D (arr1, n, m)
98     print(" B is")
99     printA2D (arr2, n, m)
100    print("N * A = ")
101    printA2D (arr3, n, m)
102    print(" -M * B = ")
103    arr4 = np.matlib.zeros ([n, m])
104    scalarMulArr (arr2, arr4, n, m, -M)
105    printA2D (arr4, n, m)
106    print("N * A - M * B = ")
107    addArr (arr4, arr3, arr3, n, m)
108    printA2D (arr3, n, m)
109    scalarCalArr (arr3, arr3, n, m, K)
110    print("N * A - M * B + K = ")
111    printA2D (arr3, n, m)
112    else:
113        break
114 main()

```

- متد **main()** ابتدا، آرایه های `arr1`، `arr2`، `arr3` و متغیر `choice` را تعریف کرده، یک حلقه تکرار `while` بی‌نهایت ایجاد می‌کند. در این حلقه اعمال زیر انجام می‌شود:
  ۱. ابتدا با فراخوانی متد `menu()` منوی را به کاربر نمایش می‌دهد.
  ۲. در این منو کاربر گزینه‌ای را انتخاب می‌کند.
  ۳. بر اساس گزینه‌ای که کاربر انتخاب می‌کند، با استفاده از یک `switch` عمل متناظر را انجام می‌دهد.
- متد `menu()` ۹ گزینه را نمایش می‌دهد و کاربر می‌تواند شماره یکی از گزینه‌ها را وارد کند. در پایان، شماره وارد شده توسط کاربر را برمی‌گرداند.
- متد `readA2D()` آرایه `arr1`، تعداد سطرها (`n`)، تعداد ستون‌ها (`m`) را به عنوان پارامتر دریافت کرده، عناصر آرایه را خوانده، در `arr1` قرار می‌دهد.
- متد `printA2D()` آرایه `arr1`، تعداد سطرها (`n`)، تعداد ستون‌ها (`m`) را به عنوان پارامتر دریافت کرده، عناصر آرایه را در خروجی نمایش می‌دهد.
- متد `addArr()` پارامترهای `arr1` (آرایه اول)، `arr2` (آرایه دوم)، `arr3` (جمع دو آرایه)، `n` (تعداد سطرها)، `m` (تعداد ستون‌ها) را دریافت کرده، حاصل جمع دو آرایه `arr1` و `arr2` را با استفاده از حلقه تودرتو حساب کرده، در آرایه `arr3` قرار می‌دهد.

## پروژه‌های برنامه‌نویسی ساده و کوچک ۲۵

- **متد subArr()** پارامترهای arr1 (آرایه اول)، arr2 (آرایه دوم)، arr3 (حاصل تفریق آرایه اول از آرایه دوم)، n (تعداد سطرها)، m (تعداد ستون‌ها) را دریافت کرده، آرایه دوم را از آرایه اول کم کرده، در آرایه سوم قرار می‌دهد.
- **متد mulArr()** پارامترهای arr1 (آرایه اول)، arr2 (آرایه دوم)، arr3 (نتیجه حاصل ضرب دو آرایه اول و دوم)، n (تعداد سطرها)، m (تعداد ستون‌ها) را دریافت کرده، با استفاده از حلقه تکرار تودرتو حاصل ضرب دو آرایه را محاسبه می‌کند.
- **متد scalarMulArr()** پارامترهای arr1 (آرایه‌ایی که باید با یک عدد value ضرب گردد)، arr2 (آرایه‌ایی که حاصل ضرب عدد آرایه اول را نگه‌داری می‌کند)، n (تعداد سطرها)، m (تعداد ستون‌ها) و value (مقداری که باید در آرایه arr1 ضرب شود) را دریافت می‌کند و با استفاده از حلقه تکرار تودرتو حاصل ضرب مقدار value را در تک‌تک عناصر arr1 محاسبه می‌کند، در arr2 قرار می‌دهد.
- **متد scalarCalArr()** پارامترهای arr1 (آرایه‌ایی که باید با عدد value جمع شود)، arr2 (آرایه‌ایی که حاصل جمع آرایه اول با عدد value را نگه‌داری می‌کند)، n (تعداد سطرها)، m (تعداد ستون‌ها) و value (مقداری که باید با تک‌تک عناصر arr1 جمع شود) را دریافت کرده، مقدار value را با عناصر آرایه arr1 جمع کرده، در آرایه arr2 قرار می‌دهد.

```
1.Read array A
2.Read array B
3.Display array A
4.Display array B
5.Add array
6.Sub array
7.Mul array
8. N * A - M * B + K
9.Exit
Select 1 to 9: 1
Enter 3 numbers for row 1 :1 2 3
Enter 3 numbers for row 2 :4 5 6
Enter 3 numbers for row 3 :7 8 9
1.Read array A
2.Read array B
3.Display array A
4.Display array B
5.Add array
6.Sub array
7.Mul array
8. N * A - M * B + K
9.Exit
Select 1 to 9: 2
Enter 3 numbers for row 1 :2 4 6
Enter 3 numbers for row 2 :8 0 1
Enter 3 numbers for row 3 :3 5 7
```

```

1.Read array A
2.Read array B
3.Display array A
4.Display array B
5.Add array
6.Sub array
7.Mul array
8. N * A - M * B + K
9.Exit
Select 1 to 9: 5
A is
  1.0   2.0   3.0
  4.0   5.0   6.0
  7.0   8.0   9.0
B is
  2.0   4.0   6.0
  8.0   0.0   1.0
  3.0   5.0   7.0
A + B =
  3.0   6.0   9.0
 12.0  5.0   7.0
 10.0  13.0  16.0

```

```

1.Read array A
2.Read array B
3.Display array A
4.Display array B
5.Add array
6.Sub array
7.Mul array
8. N * A - M * B + K
9.Exit
Select 1 to 9: 8
Enter N, M, K:2 4 8
A is
  1.0   2.0   3.0
  4.0   5.0   6.0
  7.0   8.0   9.0
B is
  2.0   4.0   6.0
  8.0   0.0   1.0
  3.0   5.0   7.0
N * A =
  2.0   4.0   6.0
  8.0  10.0  12.0
 14.0  16.0  18.0
-M * B =
 -8.0  -16.0 -24.0
 -32.0  -0.0  -4.0
 -12.0 -20.0 -28.0
N * A - M * B =
 -6.0  -12.0 -18.0
 -24.0  10.0   8.0

```

پروژه‌های برنامه‌نویسی ساده و کوچک ۲۷

```
2.0    -4.0    -10.0
N * A - M * B + K =
2.0    -4.0    -10.0
-16.0   18.0   16.0
10.0    4.0    -2.0
```

- 1.Read array A
  - 2.Read array B
  - 3.Display array A
  - 4.Display array B
  - 5.Add array
  - 6.Sub array
  - 7.Mul array
  8.  $N * A - M * B + K$
  - 9.Exit
- Select 1 to 9: 9

## ۹-۱. پیاده‌سازی برنامه انجام عملیاتی از قبیل بالا مثلثی، پایین مثلثی و غیره بر روی آرایه با توابع

برنامه‌ای که عناصر یک ماتریس سه در سه را خوانده، اعمال زیر را انجام می‌دهد:

۱. تعیین می‌کند ماتریس بالا مثلثی است یا نه؟
۲. تعیین می‌کند ماتریس پایین مثلثی است یا نه؟
۳. تعیین می‌کند ماتریس بالا و پایین مثلثی است یا نه؟
۴. تعیین می‌کند که آیا تمام عناصر هر سطر ماتریس برابرند یا نه؟
۵. تعیین می‌کند که تمام عناصر هر ستون ماتریس برابرند یا نه؟

```
1 import numpy.matlib
2 import numpy as np
3 def readA2D (a, n, m):
4     for i in range(0, n):
5         print("Enter ", n, "numbers for row ", i + 1, end = " :")
6         s = input().split(' ')
7         for j in range(0, n):
8             a[i, j] = (s[j])
9 def printA2D(a, n, m):
10    for i in range (0, n):
11        for j in range (0, n):
12            print(" ", a[i, j], end = '')
13        print()
14 def isUpTri (a, n, m):
15    for i in range(0, n):
16        for j in range(i + 1, m):
17            if a[i, j] != 0:
18                return False
19    return True
20 def isDownTri (a, n, m):
```

```

21     for i in range(0, n):
22         for j in range(0, i - 1):
23             if a[i, j] != 0:
24                 return False
25     return True
26 def isEqualRow (a, n, m):
27     for i in range(0, n):
28         for j in range(0, m):
29             if a[i, j] != a[i, 0]:
30                 return i
31     return -1
32 def isEqualColumn (a, n, m):
33     for j in range(0, m):
34         for i in range(0, n):
35             if a[i, j] != a[0, j]:
36                 return j
37     return -1
38 def main():
39     n = 3; m = 3
40     arr1 = np.matlib.zeros ([n, m])
41     readA2D (arr1, n, m);
42     printA2D (arr1, n, m);
43     if isUpTri (arr1, n, m) == True:
44         print("Upper Tringle")
45     else:
46         print("not Upper Tringle")
47     if isDownTri (arr1, n, m) == True:
48         print("Down Tringle")
49     else:
50         print("not Down Tringle")
51     if isDownTri (arr1, n, m)==True and isUpTri (arr1, n, m)==True:
52         print("Down and up Tringle")
53     else:
54         print("not Down and Up Tringle")
55     res = isEqualRow (arr1, n, m)
56     if res == -1:
57         print("Equal Each Row")
58     else:
59         print("not Equal row ", (res + 1))
60     res = isEqualColumn (arr1, n, m)
61     if res == -1:
62         print("Equal Each Column")
63     else:
64         print("not Equal Column ", res + 1)
65 main()

```

## پروژه‌های برنامه‌نویسی ساده و کوچک ۲۹

- **متد main()**، ابتدا آرایه arr با n سطر و m ستون تعریف کرده، با فراخوانی متد readA2D() عناصر آرایه را می‌خواند و با فراخوانی متد printA2D() عناصر آرایه خوانده‌شده را نمایش می‌دهد. در ادامه، با فراخوانی هر یک از متدها isUPTri() isDownTri() isEqualRow() isEqualColumn() پیغام مناسب نمایش می‌دهد.
- **متد readA2D()**، پارامترهای arr (آرایه‌ای دوبعدی اعداد خوانده را نگه‌داری می‌کند)، n (تعداد سطرها) و m (تعداد ستون‌ها) را دریافت کرده، داده‌ها را از ورودی خوانده، در این آرایه قرار می‌دهد.
- **متد printA2D()**، پارامترهای arr (آرایه‌ای که عناصر آن باید چاپ شود)، n (تعداد سطرها) و m (تعداد ستون‌ها) را دریافت کرده، عناصر آرایه arr را به خروجی می‌برد.
- **متد isUPTri()** آرایه a (آرایه‌ای که باید تعیین شود، بالا مثلثی است یا نه)، n (تعداد سطرها) و m (تعداد ستون‌ها) را دریافت کرده، اگر هر یک از عناصر که z (شماره ستون) شماره آن‌ها (j) بزرگ‌تر از  $i + 1$  باشد و مخالف صفر باشد، متد False را برمی‌گرداند (یعنی، ماتریس بالا مثلثی نیست). اگر حلقه پیمایش شود، متد True را برگشت می‌دهد (یعنی، ماتریس بالا مثلثی است).
- **متد isDownTri()** آرایه a (آرایه‌ای که باید تعیین شود، که آیا پایین مثلثی است یا خیر)، n (تعداد سطرها) و m (تعداد ستون‌ها) را به‌عنوان پارامتر دریافت کرده، اگر هر یک از عناصر که شماره ستون آن‌ها کوچک‌تر از ۱ باشد و مخالف صفر باشد، متد False را برمی‌گرداند (یعنی، ماتریس پایین مثلثی نیست). اگر حلقه‌ها خاتمه یابند، متد True را برمی‌گرداند (یعنی، ماتریس پایین مثلثی است).
- **متد isEqualRow()** آرایه a (آرایه‌ای که باید تعیین شود، آیا همه عناصر هر سطر آن با هم برابرند یا نه)، n (تعداد سطرها) و m (تعداد ستون‌ها) را به‌عنوان پارامتر دریافت کرده، شماره اولین سطری که همه عناصر آن برابر نباشند، را برمی‌گرداند. اگر همه سطرها، همه عناصرشان برابر باشند، متد -۱ را برگشت می‌دهد.
- **متد isEqualColumn()** آرایه a (آرایه‌ای که باید تعیین شود، آیا همه عناصر هر ستون آن با هم برابرند یا نه)، n (تعداد سطرها) و m (تعداد ستون‌ها) را به‌عنوان پارامتر دریافت کرده، شماره اولین ستونی که همه عناصر آن برابر نباشند، را برمی‌گرداند. اگر همه ستون‌ها، همه عناصرشان برابر باشند، متد -۱ را برگشت می‌دهد.

```

Enter 3 numbers for row 1 :1 0 0
Enter 3 numbers for row 2 :1 1 0
Enter 3 numbers for row 3 :1 1 1
    1.0    0.0    0.0
    1.0    1.0    0.0
    1.0    1.0    1.0
Upper Tringle
    
```