مهراد واعظی(۹۵۲۳۱۲٤) گزارشکار تمرین سری ۵

سوال۱)

ابتدا دو وکتور vec1 و vec2 را با سایز های به ترتیب ۱۰۰ و ۱۰ میسازیم. سپس یک لامبدا فانکشن (allocate) میسازیم که با استفاده از متغیر لوکال i ورودی را با i جمع میکند و در هر مرحله i را یکی اضافه میکند. حال با استفاده از الگوریتم std::for_each() از اول تا آخر وکتور را طی کرده و هر درایه را با i جمع کرده و i را در هر مرحله یکی اضافه میکنیم. سپس این کار را دوباره برای vec2 انجام میدهیم. بدین ترتیب هر دو وکتور پر میشوند. برای چاپ وکتور ها نیز با همین الگوریتم، لامبدا فانکشن را بصورت cout برای هر درایه مینویسیم. برای کپی کردن vec1 در آخر vec2 با الگوریتم std::copy() از اوّل تا آخرِ vec1 را به آخر vec2 اضافه میکنیم. برای استخراج اعداد فرد با اسفاده از الگوریتم std::copy_if() شرط فرد بودنِ عدد را توسط لامبدا فانکشن اضافه کرده و اعدادی را که در شرط صدق کنند به آخر وکتورِ odd_vec اضافه میکنیم. برای ذخیره کردن vec1 بصورت برعکس داخل وکتور reverse_vec از الگوریتم std::reverse_copy() استفاده میکنیم.

توجّه: کتابخانه ی <execution> از قرار معلوم در ورژن های فعلی C++17 موجود نمیباشد. (با ۲ کامپایلر g++ و myngw تست شد!) بدین جهت کدِ مربوط به این بخش نوشته شد و بصورت کامنت قرار داده شد. در صورت ران شدن در کامپیوترِ مصحّح این بخش و include<execution> از حالت کامنت خارج شود.

مهراد واعظی(۹۵۲۳۱۲٤) گزارشکار تمرین سری ۵

سوال ۲)

وقتی آرگومانی بصورت args* وارد میشود یعنی آن تابع، یک tuple را از ورودی میگیرد. همچنین آرگومانی بصورت keywords** نشان میدهدکه آن تابع یک dictionary را از ورودی میگیرد. این تابع بدین صورت عمل میکندکه به تعداد ورودی های عادی خود ورودی ها را گرفته و سایر ورودی ها را بصورت tuple در args ذخیره میکند و اگر داده ای بصورت (key, keyword) pair داده شود آنها را نیز در keywords ذخیره میکند. مثال:

```
def func (a, b, *args, **keywords):
    print(f{‘ A = {a} and B = {b}’})
    for arg in args:
        print(arg, ” ,  “)
    for kw in keywords:
        print(kw, “ : ”, keywords[kw])
```

حال این تابع را بصورت زیر فراخوانی میکنیم:

```
func(“Number_1”, “Number_2”, “Number_3”, “Number_4”, “Integer”, Student_1=”John”,
    Student_2=”Paul”, Student_3=”Guido”)
```

خروجی به شکل زیر خواهد بود:

```
A = Number_1 and B = Number_2
Number_3,  Number_4,   Integer,
Student_1 : John
Student_2 : Paul
Student_3 : Guido
```

مشاهده میشود که ۲ آرگومان اوّل به عنوان a و b گرفته شده اند بقیه ی آرگومان ها بصورت tuple گرفته شده اند و آرگومان هایی که بصورت دیکشنری وارد شده اند تشکیل یک دیکشنری در keywords را داده اند. فایل این برنامه در فولدر Q2 قرار داده شده است.

مهراد واعظی(۹۵۲۳۱۲٤) گزارشکار تمرین سری ۵

سوال ۳)

الف)

A0: تابع zip() ۲ tuple بصورت ورودی گرفته و درایه های آن دو را بصورت یک به یک بصورت یک ماتریس ۵ در ۲ وارد تابع dict() میکند. این تابع نیز این ماتریس را گرفته و آن را تبدیل به یک دیکشنری ۵ تایی میکندکه کلیدهای آن حروف و مقادیر آن ها اعداد هستند.

A1: اعداد ۰ تا ۹ را وارد بصورت range وارد A1 میکند.

A2: i متشکل از اعداد ۰ تا ۹ میباشد. امّا کلید های A0 تنها متشکل از 'a', 'b', 'c', 'd', 'e' هستند پس i جزو هیچیک نمیباشد و A2 درکل یک لیست خالی است.

A3: i متشکل ازکلیدهای A0 است پس A0[i] مقادیر این کلید ها را میدهدکه بصورت صعودی sort شده اند و تابع sorted() تغییری ایجاد نمیکند.

A4: i در اینجا متشکل از اعداد ۰ تا ۹ است پس A4 لیستی متشکل از لیست های ۱ در ۲ ای است که درایه ی اوّل اعداد ۰ تا ۹ و درایه ی دوّم مربع این اعداد است.

ب)

جهت این کار ابتدا با استفاده از متغیر globals() تمام متغیر هایی که با 'A' شروع میشوند را داخل لیستی میریزیم و سپس با استفاده از این کلید ها مقادیر این متغیر ها را از globals() در می آوریم. سپس در یک حلقه این مقادیر را نمایش میدهیم.

سوال ٤)

اصلاحیه: حاصل تقسیم تعداد نقاط داخل دایره بر نقاط کل برابر π/٤ است که در صورت سوال π قید شده. بدین جهت نتیجه ی تقریب (P) را ضربدر ٤ میکنیم.
تابع IsInCircle(x, y) را طبق فاصله ی نقطه از مرکز تعیین میکنیم. در صورت کمتر بودن ار ۰/۵ نقطه داخل دایره است.
تابع Find() بدین صورت عمل میکندکه تا موقعی که اختلاف تقریب(P) از خود عدد پی کمتر از ۰/۰۱ نشده حلقه را ادامه میدهد و وقتی چنین شرطی برقرار نشد حداقل تعداد نقاط را چاپ میکند و سپس مقدار تقریب زده شده ی عدد پی را برمیگرداند. سپس تعداد تکرار تابع Find() را ازکاربرگرفته و میانگین اعداد پی برگردانده شده از Find() را چاپ میکند.

مهراد واعظی(۹۵۲۳۱۲٤) گزارشکار تمرین سری ۵

سوال ۵)

برای استفاده از این فایل بایستی آن را import کنید. از ماژول pathlib کلاس Path اضافه شده و ماژول os نیز اضافه شده است. تابع create_dir() ابتدا بررسی میکندکه فولدری با نام name و آدرسِ address موجود نباشد و سپس آن را تشکیل میدهد وگرنه اخطار میدهدکه چنین فولدری از قبل موجود میباشد. تابعِ create_file() نیز تست وجود فایل را انجام میدهد. **دقّت** شود که در بخشِ name نام فایل همراه با نوع آن فایل (مثلا 'test.txt' یا 'test.pdf') وارد شود. تابعِ find() نیز با استفاده از دستور Path.glob() تمام فولدر و زیر فولدر هایی که شامل فایلی هستندکه نام آن فایل بصورت دقیق بدون هیچ پسوند و پیشوندی برابر {name} باشد را بر میگرداند. لذا **باید** نام فایل بصورت کامل و همراه با نوع فایل (مثلِ 'main.cpp' یا 'test.txt') وارد شود. تابعِ delete() در صورت وجود فایلی با نام name در آدرسِ address آن را پاک میکند. تابعِ delete_dir() نیز در صورت وجو فولدری با نام و آدرسِ قید شده آن را پاک میکند.

سوال ٦)

فایل IntegrateC++.exe ضمیمه شده که در صورت نیاز به اجرا بایستی آن را با comamnd و عددی به عنوان درجه (n) اجرا کرد که البتّه نیازی نخواهد بود. کلاسِ PGaussSolver که در ماژول PgaussSolver.py قرار دارد مشابه کد C++ نوشته شده است. ماژول های math و subprocess (برای اجرای IntegrateC++.exe) و time و matplotlib.pyplot و numpy و PgaussSolver (به عنوان PyGauss) اضافه شدند. تابعِ aFunction() مطابق تابعِ قید شده در صورت سوال تعریف شده است. لیست های P_time و C_time جهت ذخیره کردنِ زمان های اجرای کد های پایتون و C++ قرار داده شده است. حال به ازای درجه های ۱ تا ۲۰ چندجمله ای های لژاندرکد های هردو زبان اجرامیشود و زمان های هردو ثبت میشود. پاسخ های هردو نیز درکامند چاپ میشود. پس از این کار ۲ لیست P_time و C_time تشکیل می یابند. لیست data بدین شکل تولید میگرددکه سطر اوّل آن اعداد ۱ تا ۲۰ که درجه(n) است و سطر های دوم و سوم به ترتیب لیست های P_time و C_time هستند. با استفاده از دستور های matplotlib جدول تشکیل شده و برای هردو زبنا در یک figure ۲ نمودارکه محور افقی n و محور عمودی زمان اجرا میباشد رسم میشود. مشاهده میشود که برای $n > 13$ زمان اجرا ی C++ به مراتب کمتر از زمان اجرای Python است که بصورت نمایی در حال افزایش است. فایل pdf شده ی این جدول و نمودار نیز ضمیمه شده است.

مهراد واعظی(۹۵۲۳۱۲٤) گزارشکار تمرین سری ۵

سوال ۷)

توجّه: چون برنامه باید تنها از ۱ خط تشکیل شده باشد برای دادن ورودی به برنامه باید آن ها را از طریق command window وارد کنید. پس خطی برای دریافت ورودی در نظرگرفته نشده است.

(مثلاً: $$ python Q7_main.py 1 2 3 4 5 6 7 8 9 10 11 12)

برای گرفتن ورودی ها از command window از ماژول sys استفاده میکنیم. ابتدا درونی ترین لیست که لیست اصلی است را تشکیل میدهیم. بدین صورت که sys.argv[i] ها را وارد لیست میکنیم با این شرط که اندیس i مضرب ٦ باشد و نیز خود sys.argv[i] نیز مضرب ٦ باشد(دقت شود که Q7_main.py برابر sys.argv[0] میباشد) سپس این string را با تابع int() به عدد تبدیل میکنیم. سپس برای اینکه اعداد تکراری را از لیست خارج کنیم از دستور set() استفاده میکنیم و در پایان با دستور sorted() این لیست را سورت کرده و نشکیل میدهیم. اما اکنون تابع print() این لیست را بصورت [٦, ۱۲] خروجی میدهد. لزا جهت چاپ آن بصورت ۱۲ ٦ باید از ' * ' استفاده کنیم تا تمام درایه ها ی آن را تکتک چاپ کند.

Git)

آدرس Repository به قرار زیر است: https://github.com/MehradVaezi/AP-HW5.git
ابتدا با دستور git init در فایل موردنظر یک فولدر git. ساخته و سپس با دستور git config –global –edit مشخصات را وارد میکنیم . سپس آدرس Repository را با دستورgit remote add origin وارد میکنیم . حال با دستور touch .gitignore این فایل را میسازیم و با دستور nano آن را بصورتی تغییر میدهیم که فایل های o.* و ~*.* را نخواند . حال با دستور ...git add Q1/ Q2/ Q3/ Q4/ این فایل ها را در حالت tracked قرار میدهیم و سپس با دستور git commit –m “AP files added” این فایل ها را commit میکنیم . سپس با دستور git push origin master آنها را در ریپو ی خودمان آپلود میکنیم.