

بسم الله الرحمن الرحيم

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

گزارش تمرین سری پنجم

برنامه نویسی پیشرفته

دکتر جهانشاهی

وحید پوراکبر

9523022

Git

https://github.com/vahid-aki/AP_HW5.git

C++

سوال 1:

{آ}: برای ایجاد وکتور بدون استفاده از حلقه از دستور `for_each` استفاده شده است به این صورت که یک `iterator` از اول وکتور تا آخر آن می رود (برای این منظور باید در ابتدا `size` وکتور را مشخص کنیم) و به هر المانی که میرسد آن را برابر `n` قرار می دهد و دفعه بعد `n` یکی اضافه میشود (بدین منظور از لاتدا فانکشن استفاده شده است) و به همین ترتیب وکتور مقدار میگیرد.

```
std::for_each(vec2.begin(), vec2.end(), [&n](int& a){a = (n++);});
```

{ب}: برای این منظور از دستور `vec.insert` استفاده شده است به این صورت که از اولین المان تا آخرین المان `vec1` به آخر `vec2` اضافه شده است (از آخرین المان `vec2` شروع به اضافه شدن کرده است).

```
vec2.insert(vec2.end(), vec1.begin(), vec1.end());
```

{ج}: برای این کار دستور `copy_if` و استفاده از لاتدا فانکشن به کار برده شده است. و چون آن المان هایی که زوج هستند را صفر قرار داده برای همین از دستور `vec.erase` برای پاک کردن آنها استفاده شده است.

{د}: به سادگی و با استفاده از دستور `reverse_copy` انجام میشود.

{ه}: این قسمت چون هنوز به کتابخانه `c++17` اضافه نشده به صورت کامنت آورده شده است.

Python

سوال 2:

اگر برای مثال داشته باشیم: `def func(a, b, c, ..., *args, **keywords):` آنگاه وقتی به تابع ورودی میدهیم تابع ورودی اول را در `a` و ورودی دوم را در `b` و ... به همین ترتیب در متغیرها قرار می دهد تا وقتی که به `*args` برسیم بعد از آن تابع متغیرها را در `*args`

که در واقع tuple میباشد قرار میدهد بنابراین بقیه متغیرها در یک tuple قرار داده میشوند تا وقتی که به متغیرهایی برسیم که به صورت مپ شده (در ++c) به تابع داده میشوند که در این صورت آنها را در keywords که در واقع dictionary (در پایتون) میباشد قرار می دهد.

مثال در فولدر Q2 آورده شده است.

سوال 3:

```
A0 = dict(zip(('a', 'b', 'c', 'd', 'e'), ('1', '2', '3', '4', '5')))    # {'a': '1', 'b': '2', 'c': '3', 'd': '4', 'e': '5'}
A1 = range(10)                                                       # range(0, 10)
A2 = [i for i in A1 if i in A0]                                       # []
A3 = sorted(A0[i] for i in A0)                                       # ['1', '2', '3', '4', '5']
A4 = [[i, i*i] for i in A1]     # [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]
```

سوال 4:

این سوال یک اشکال دارد و آن هم اینکه در صورت سوال گفته شده نسبت تعداد نقاط داخل دایره به تعداد کل نقاط برابر عدد پی است که اشتباه است چون این عدد قطعا کوچکتر از 1 میباشد. نسبت تعداد نقاط داخل دایره به تعداد کل نقاط برابر یک چهارم عدد پی می باشد و این سوال با این نسبت حل شده است.

سوال 5:

در این سوال دقیقا مشخص نشده که ورودی ها باید از کجا وارد شوند برای همین به صورت پیش فرض address محلی است که برنامه در آن قرار دارد و اجرا میشود با استفاده از دستور os.getcwd() و name یک فایل تکست خالی با نام test.txt انتخاب شده است.

توجه: برای نشان دادن درستی اجرای کد نیز یک پوشه با نام new folder ایجاد سپس یک فایل با نام test.txt داخل آن ایجاد میشود سپس با فراخوانی تابع find و پیدا کردن فایل test.txt در آخر همین فایل از فولدر new folder حذف میشود.

اگر قرار باشد که آدرس و اسم فایل و ... از کاربر گرفته شود آنگاه از input استفاده میکنیم که در کد کامنت شده است.

سوال 6:

همانند کد ++c این کد نیز با استفاده از کلاس ها نوشته شده است و کد آن بسیار شبیه کد ++c میباشد اما همانطور که دیده میشود زمان اجرای کد در ++c به مراتب کمتر از زمان اجرا در پایتون میباشد.

نکته: زمانی که کد در محیط ویندوز اجرا میشود زمان اجرا ها به خصوص زمان اجرای کد ++c درست نمایش داده نمیشود و اکثر مواقع صفر نشان میدهد و بعضی مواقع نیز زمان اجرای کد پایتون نیز صفر نشان میدهد برای همین وقتی کد را چند بار اجرا میکنیم برخی مواقع درست و اکثر مواقع اشتباه نشان میدهد پس بهتر است در محیط linux اجرا شود.

سوال 7:

این سوال با استفاده از چندین حلقه `for` نوشته شده است در حلقه `for` اول، اعداد ورودی داخل یک دیکشنری (`dict1`) قرار داده میشوند. سپس در حلقه `for` دوم اعداد خاص پیدا شده و داخل `dict` قرار داده می شوند. سپس با استفاده از دستور `sort` اعداد به صورت صعودی مرتب میشوند. در حلقه سوم اعداد تکراری از `dict` حذف شده و داخل یک `list` قرار داده میشوند. برای چاپ خروجی دوباره با استفاده از یک حلقه اعداد به صورت استرینگ و گذاشتن یک فاصله در `output_nums` قرار داده میشوند.