**آزمایش 3**

فاطمه راق 40013045

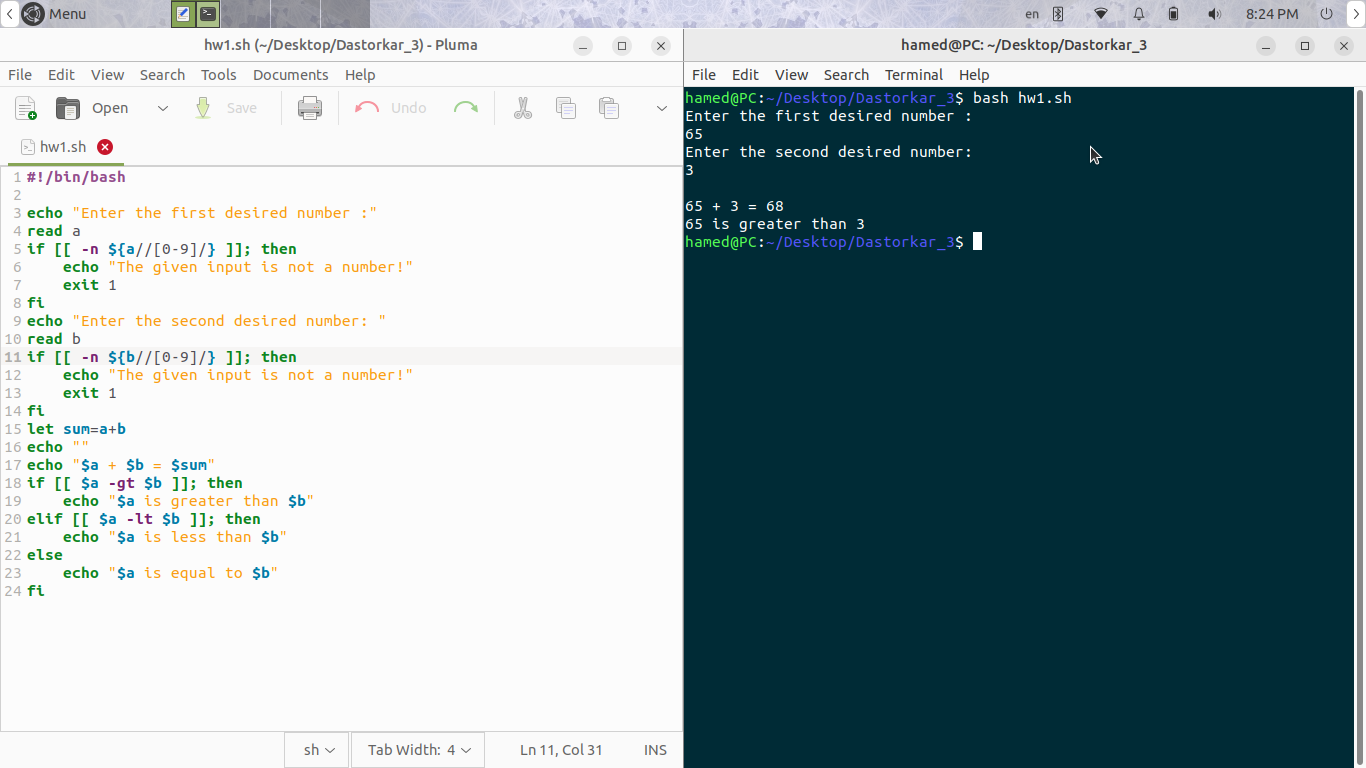
حامد فلاحی 9931043

***خلاصه***

بطور کلی، bash مفسر زبان دستورات است. Shell یک پردازشگر ماکرو است که دستورات را اجرا میکند. اصطلاح پردازشگر ماکرو به معنای عملکرد است که در آن متن و نمادها برای ایجاد عبارات بزرگتر گسترش میابد. Unix shell یک زبان برنامه نویسی و مترجم دستورات است که ویژگی های این زبان برنامه نویسی اجازه میدهد که مجموعه ای غنی از سرویس های GNU ترکیب شوند. فایل هایی که حاوی دستورات هستند میتوانند ایجاد شوند و خودشان تبدیل به دستور شوند. حال ما قصد داریم در این آزمایش، باتوجه به نحوه کار با این زبان برنامه نویسی و ویژگی های آن که در دستورکار نیز آمده، تمرین های گفته شده را مورد بررسی قرار داده و آن هارا حل کنیم.

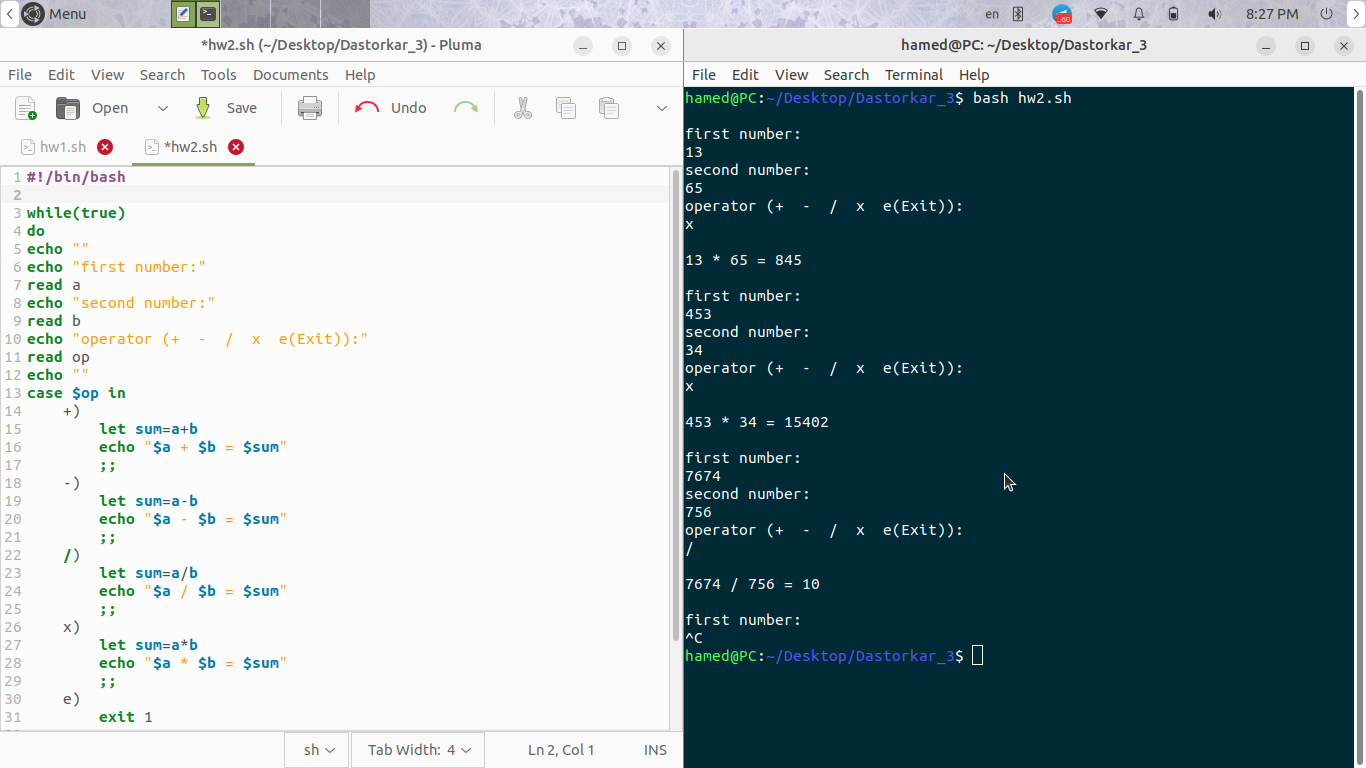
**HW1)**

در تمرین 1، از ما خواسته شده که دو عدد بعنوان ورودی به برنامه بدهیم و سپس جمع آن دو عدد را بهمراه مقایسه دو عدد به ما بعنوان خروجی بدهد. همچنین درصورت اشتباه، پیغام مناسب برای ما چاپ کند. همانطور که در عکس میبینیم، با استفاده از حالت شرطی که در استورکار نیز گفته شده، برای a و b هرکدام read مینویسیم و به ازای آنها شرط میگذاریم تا پیغام مناسب به ما بدهد. همچنین از دستور echo برای چاپ و ویزگی های دیگر نیز همانند عکس استفاده میکنیم. سپس با استفاده از دستور let مقدار جمع اعداد را در متغیر تعریف شده sum میریزیم و پس از آن با استفاده از ویژگی های شرطی، مقایسه را بین a و b انجام میدهیم. همچنین با استفاده از دستور bash (file) برنامه را اجزا کرده و نتایج مورد نظر را همانند سمت راست عکس زیر میبینیم.



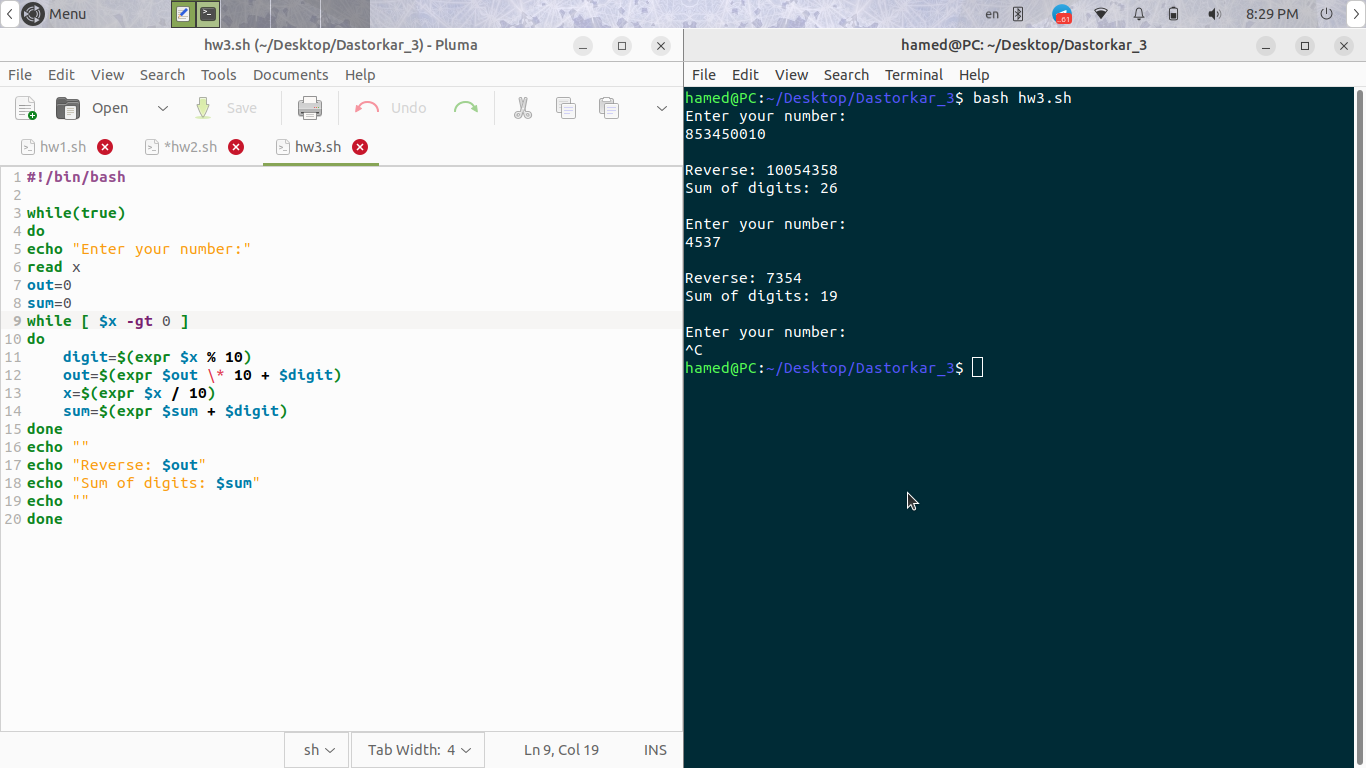
**HW2)**

در تمرین 2 از ما میخواهند که ماشین حساب ساده ای با ویژگی case بسازیم. در این تمرین نیز از ویژگی هایی که در تمرین 1 استفاده شد، استفاده میکنیم. میتوانستیم در ابتدای کد از ویژگی حلقه استفاده نکنیم و دائما عملیات را در حلقه نگه نداریم... اما این کار را بر فرض انجام دادیم و مشکلی ندارد. سپس ورودی های a و b و ورودی عملگر op را میخوانیم تا عملیات را با استفاده از آن روی هردو عدد آرگومان اعمال کنیم. همانطور که میبینیم، پس از اعمال حالت ها، درحالت ضرب، از حرف اینگلیسی x استفاده شده. دلیلش این است که عملگر ضرب \* در این زبان برنامه نویسی در اینجا به معنای هرچیزی غیر از کلید ها میباشد(همانند else) و ما این کار را برای حل این مشکل انجام دادیم و از آن نتیجه گرفتیم. همانطور که میبینید، خروجی این کد نیز به ازای اکثر عملگر ها نمایش داده شده است. همچنین حواسمان باشد که خروجی عملیات تقسیم، یک عدد آرگومان و صحیح است و به عدد پایینتر گرد میشود.(مثلا 64 🡪 64.843)



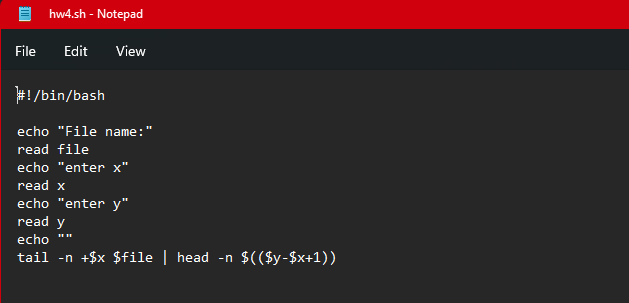
**HW3)**

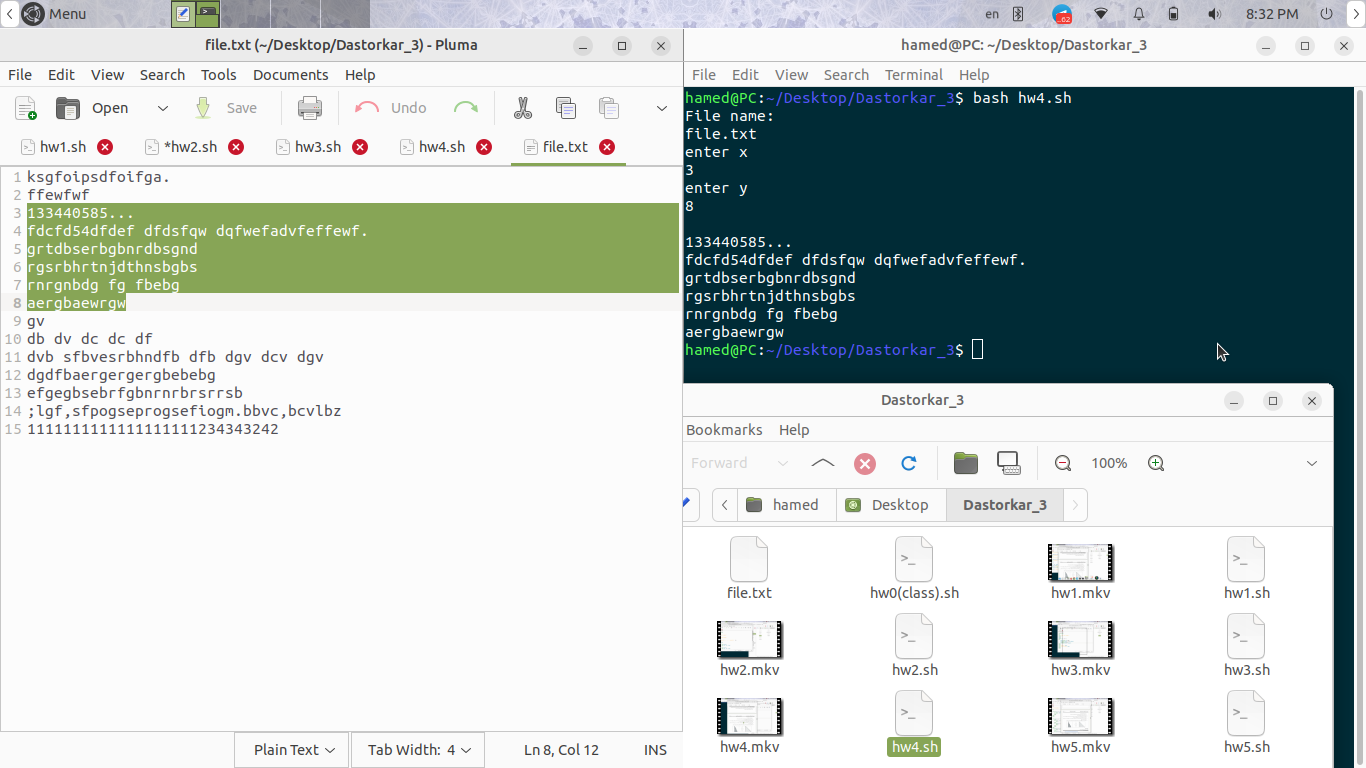
در این تمرین به ما گفته شده که برنامه نوشته شده طوری باشد که ورودی یک عدد چند رقمی و خروجی آن عدد صحیح را برای ما معکوس کند و پس از آن، جمع تک تک ارقام چند رقمی را برای ما حساب کند. همانطور که میبینید، ما برای معکوس کردن نیاز به عملگر باقی مانده و تقسیم و همچنین ویژگی حلقه داریم تا به تعداد ارقام این عدد، در حلقه بچرخد و عملیات لازم را انجام دهد. عملیات از این قرار است که ابتدا باقیمانده آن عدد به 10 گرفته شود تا تک تک ارقام را به ازای هر حلقه بدست آوریم. سپس چون خروجی همیشه با تک رقم جدا شده جمع میشود، ارقام کم ارزش باید تبدیل به بیشترین ارزش شوند و به همین دلیل به ازای هر حلقه در 10 ضرب میکنند و با رقم جدید جمع میکنند تا اولویت آن به ازای هر حلقه بالاتر رود. پس از آن نیز عدد ورودی تقسیم بر 10 میشود تا یک واحد از کم ارزش ها کم شود و رقم بعدی جلو آید و پس از آن به ازای هر حلقه، رقم هارا جمع میکند. همانطور که میبینیم، در سمت راست عکس، نتایج مورد نظر میبینیم و میبینیم درست اجرا میشود و خروجی مورد نظر را دریافت میکنیم.



**HW4)**

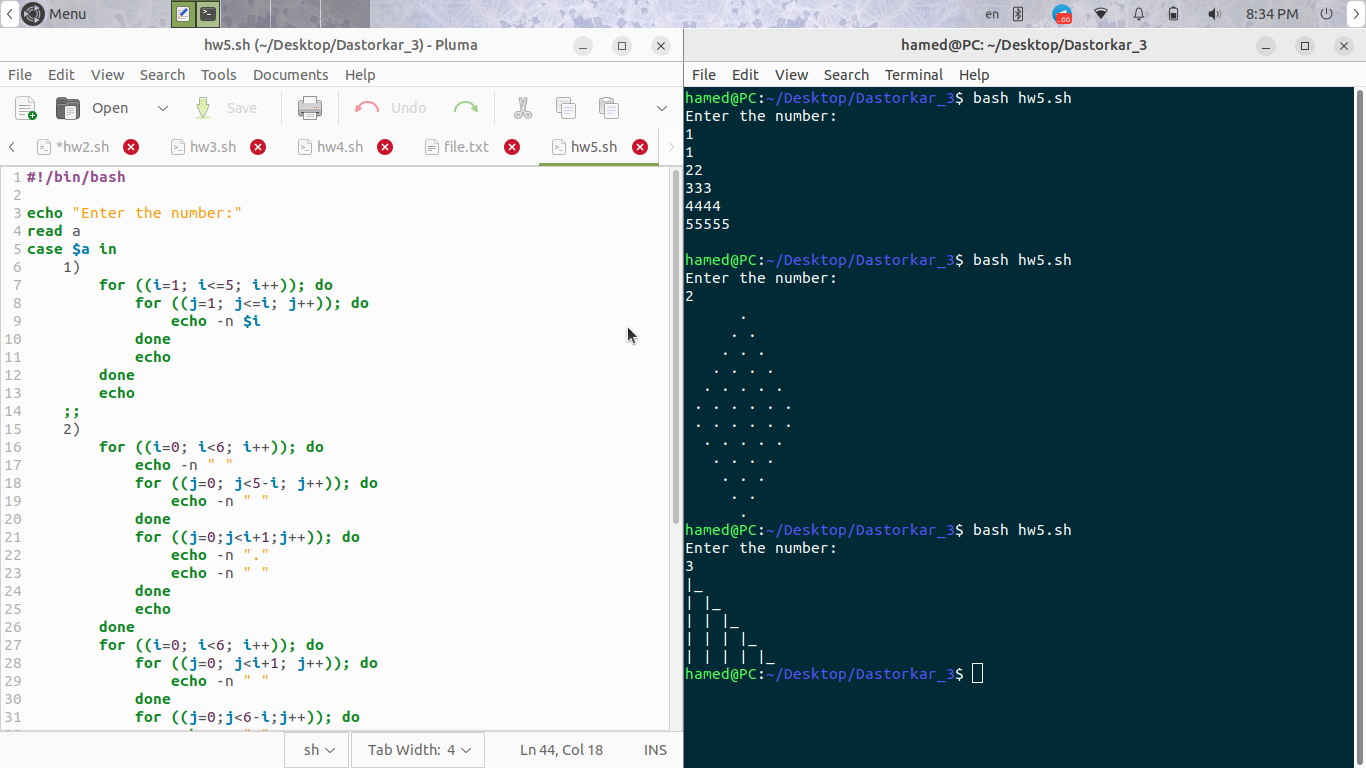
در این قسمت نیز از ما میخواهد که یک فایل را بعنوان ورودی بدهیم و سپس از خط x تا خط y که میخواهیم برایمان چاپ شود را ورودی میدهیم. همانطور که میبینیم، در این کد از دستورات tail و head استفاده شده. وظیفه این دو دستور به ترتیب اینگونه است که باتوجه به اسم فایلی که دریافت میکنند، از خط x به تعداد (y-x+1) خطوط نوشته فایل را به ما نمایش میدهند و در نهایت خروجی را در عکس مشاهده میکنیم که خطوط نمایش داده شده در ترمینال، همانند نوشته موجود در فایل میباشد.

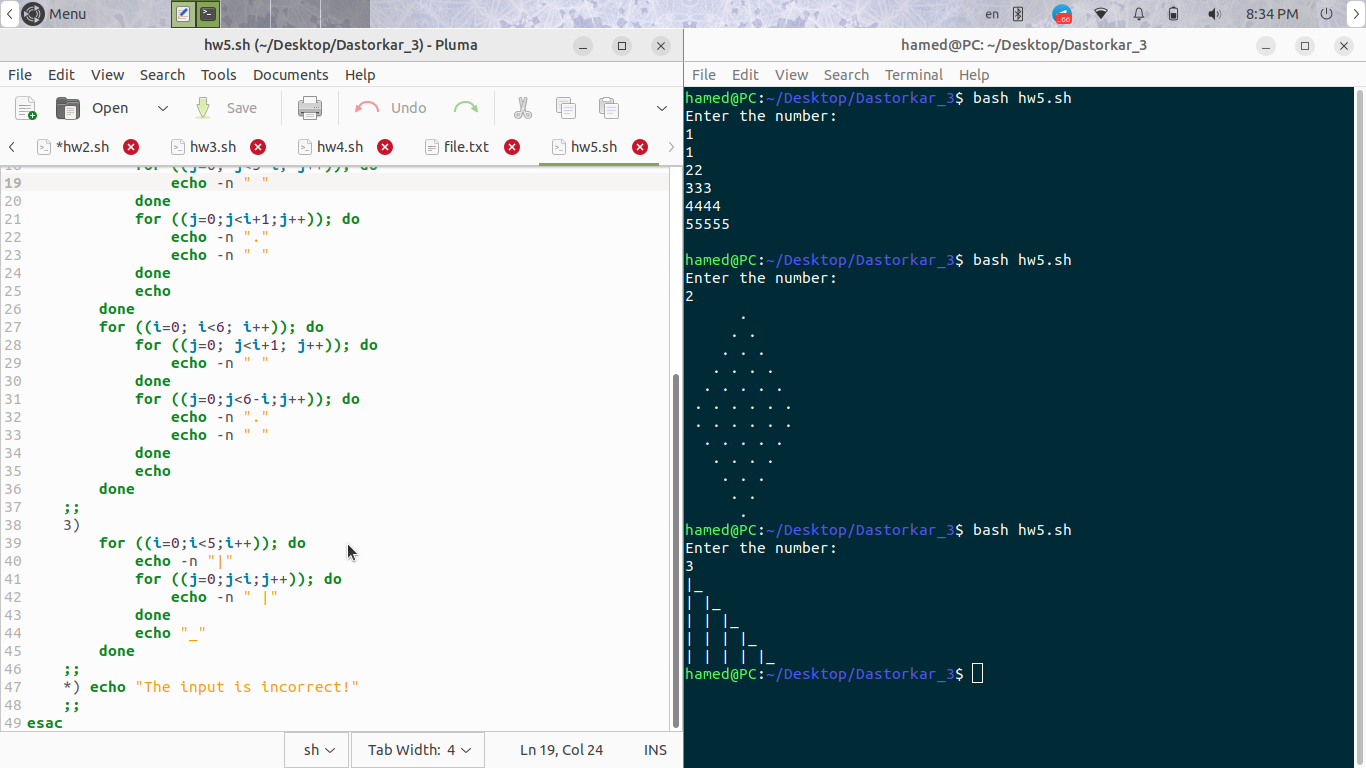




**HW5)**

در این تمرین از ما میخواهد که باتوجه به اشکال داده شده به ترتیب از عدد 1 تا 3 که بعنوان ورودی میدهیم، این اشکال را برای ما چاپ کند. ابتدا باید از تکنین case استفاده کنیم. سپس به ازای عدد 1 و شکل 1، ما باید از 2 حلقه تو در تو استفاده کنیم تا باتوجه به اینکه در هر خط یک عدد اضافه میشود، به ازای i در هر خط، کمتر از شماره خط، خود i را چاپ کنیم. در شکل 2 و عدد 2، ما باید این لوزی را به 2 مثلث برعکس تقسیم کرده و یک الگوریتم برای یک مثلث ایجاد کنیم. همچنین میبینیم کهدرون یک حلقه، 2 حلقه وجود دارد که حلقه اولی، تعداد جای خالی و حلقه دومی، تعداد نقطه هارا چاپ میکند. پس از ایجاد یک مثلث، این الگوریتم را به کلی کپی کرده و 2 حلقه درون حلقه اصلی را باهم جابجا میکنیم و با کمی تصحیح جای مثلث، نتیجه صحیح را بدست می آوریم. در شکل 3 و عدد 3، الگوریتم کاملا شبیه الگوریتم شکل 1 میباشد. با این تفاوت که بجای i، علامت | پرینت میشود و انتهای حلقه از \_ استفاده میکنیم و نتیجه صحیح را بدست می آوریم. همانطور که در خروجی ترمینال میبینیم، نتایج کاملا صحیح و درستی به ما بعنوان خروجی میدهد.





**6HW)** (امتیازی)

در این تمرین همانطور که در ویدیو گفته شد، باید ماشین حسابی درست کنیم که به ترتیب تعداد اعداد دریافتی را وارد کرده. سپس اعداد را وارد میکنیم. در مرحله بعد تعداد عملگرها را وارد کرده. سپس خود عملگر هارا وارد میکنیم(حواسمان باشد که چون عملگرها به ترتیب میان اعداد قرار میگیرند، پس معمولا تعداد عملگرهای وارد شده برابر با تعداد اعداد منهای 1 میباشد). حالا چون که باید یک ماشین حساب باشد، پس اولویت انجام عملیاتها با ضرب و تقسیم هاست. به همین دلیل اول حلقه برای ضرب و تقسیم گذاشته. بعد از اتمام آن حلقه، حلقه ای دیگر برای جمع و تفریق میگذاریم و نتیجه هارا با result جمع میکنیم. نتیجه ران شده به شکل زیر است:

