

سوال اول : ابتدا ابعاد نقشه از ورودی گرفته میشود و تابع rand را برای ساختن اعداد رندم فراخوانی می کنیم تا با هربار اجرا شدن برنامه اعداد متفاوت ساخته شود و یک object از کلاس map با ورودی n میسازیم. و به ترتیب توابع showmap و (قسمت اول) showRoute و (قسمت دوم) showRoute_b و (قسمت سوم) showbest فراخوانی میشوند.

کلاس map : برای ساختن یک object یک عدد به عنوان ورودی میگیرد و ارایه ای داینامیک دوبعدی map به آن ابعاد میسازد و با اعداد رندوم بین صفر تا صد پر میکند .

*ارایه های مسیر هر قسمت اعداد یک مسیر طی شده را نشان میدهد.

قسمت اول : در تابع showroute تابع findroute فراخوانی میشود و سپس ارایه route چاپ میشود .

در تابع findroute در یک حلقه while از مبدا یعنی (0,0) شروع کرده تا زمانی که به خانه (n-1,n-1) نرسیده ایم ، در هر مرحله چک میشود که بین اختلاف ارتفاع با خانه سمت راست و پایین کدام کوچک تر است و مسیر بهتر انتخاب شده و مکان فعلی ابدیت شده ، میزان اختلاف با اختلاف های قبلی جمع شده و خانه مربوطه در ارایه route مقدار یک میگیرد . در هر مرحله چک میشود که اگر به اخر ارایه رسیده بودیم تنها مسیری که برایمان باقی مانده طی شود . در انتها جمع اختلاف ها را چاپ میکند.

قسمت دوم : در تابع showroute_b تابع findroute_b فراخوانی میشود و سپس ارایه route_b چاپ میشود .

همانند قسمت اول عمل میشود با این تفاوت که در هر مرحله راه قطری هم چک شده و اگر بهترین بود به آن می رود.

قسمت سوم : در تابع showbest تابع findbest فراخوانی شده و bestroute چاپ میشود .

تابع findbest : برای رسیدن به خانه اخر پایین سمت راست و شروع از خانه بالا سمت چپ در صورتی که فقط پایین و راست می توان رفت ، نیاز به n-1 حرکت به سمت راست و n-1 حرکت به سمت پایین داریم . برای اینکار در v vector اعداد 2 و 3 به ترتیب نماینده سمت راست و پایین هستند را به تعداد مناسب وارد می کنیم و با کمک کتابخانه algorithm تمام جایگشت های آن را که نشان گر راه های ممکن است چک می کنیم . در ابتدا متغیر های sum_min و v1 vector که متغیر های کمکی که در هر مرحله جمع اختلاف میمنیم و بهترین مسیر تا آن لحظه را در خود نگه میدارند استفاده میکنیم.

در یک حلقه do-while تا زمانی که جایگشتی باقی مانده به صورت عمل می کنیم که به ازای عدد 2 به سمت راست و عدد 3 به سمت چپ می رویم و جمع اختلاف ها را در sum ذخیره کرده و در انتها با sum_min مقایسه کرد می کنیم اگر کوچک تر بود مقدار sum_min را ابدت کرده و مسیر ذخیره شده در وکتور را ابدیت می کنیم (ابتدا وکتور را خالی کرده و سپس در آن مسیر جدید را ذخیره می کنیم)

پس از تمام شدن این حلقه ، مسیر نهایی در ارایه bestroute مشخص کرده و مقدار sum_min را چاپ می کنیم .

سوال دوم) در این سوال کلاس های خواسته شده که شامل یک تابع `counter` هستند تعریف می شود .

تابع `runtime` به صورت `Template` تعریف شده که ورودی آن یک تابع از یک کلاس (ورودی تابع `int` و خروجی آن `int`) و یک عدد `int n` دیگر است . که در آن یک `object` از کلاسی که تابع آن فرستاده شده ساخته میشود و تابع با ورودی `n` بین دو تابع از کتابخانه `chrono` فراخوانی شده تا زمان آن محاسبه شده و به میکرو ثانیه بازگردانده شود.

در `main` در یک حلقه به با `n` های متفاوت و پوینتر هایی به توابع کلاس ، تابع `runtime` فراخوانی شده و زمان آن چاپ میشود .

سوال سوم) دو کلاس `db` و `dbnew` با متغیر های مناسب تعریف شده اند .

در ابتدا فایل `db.txt` با می کنیم و خط اول آن که تعداد تراکنش ها نوشته شده است را می خوانیم و ذخیره می کنیم .

یک ارایه از جنس کلاس `db` به طول تعداد تراکنش ها می سازیم و در یک حلقه `for` یک خط از فایل را خوانده و در متغیر های کلاس ذخیره می کنیم .

یک `object` از کلاس `dbnew` میسازیم و تاریخ اولین المان از ارایه `db` را در آن ذخیره می کنیم . در یک حلقه فور چک میشود اگر تاریخ با تاریخ `object dbnew` یکی باشد ، چک میشود که در وکتور شماره کالا و وکتور شماره مشتری اگر در مشخصات موجود نباشد به آن ها اضافه میشود .

و اگر تاریخ یکی نباشد اندازه وکتور شماره کالا ها و شماره مشتری به عنوان تعداد مشتری ها و تعداد کالا در `dbnew` ذخیره شده ، در فایل `dbnwe.txt` ذخیره شده .

وکتور ها خالی و تاریخ ابدیت شده و مشخصات جدید در آن ذخیره شده و ادامه پیدا می کند.

تمرین کارگاه) در قسمت `head` فایل های `css` اد شده اند و در قسمت `body` در یک `container` قسمت های بالا و `contact` ها و قسمت پیام ها قرار دارند که در هر کدام باکس ها و `icon` های مناسب قرار داده شده است و `style` مربوط به هر کدام از بخش ها که با کلاس های متفاوت نام گذاری شده اند و در فایل `style.css` نوشته شده است .

برای قرار دادن کد ها در `github` بعد از ساختن `repository` در جایی که فایل ها قرار دارد یک فایل `.gitignore` را به این صورت که بعد از `**.*` و بقیه فایل هایی که نمیخواهیم به همین ترتیب نوشته شده باشد . و با دستورات زیر کد ها را در آن قرار می دهیم

```
Git init
git remote add origin https://github.com/MahboobeSh/AP-HW2.git
git push -u origin master
```