

National College of Ireland

Project Submission Sheet

Student Name:AMALACHUKWU ADAEZE ATUSIUBA.....

Student ID:x23293012.....

Programme: MSC AI..... **Year:**2024/2025...

Module:Programming for Artificial Intelligence (MSCAI1).....

Lecturer:Shreyas Setlur Arun.....

Submission Due Date:10/12/2024.....

Project Title:

Word Count:

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:ADAEZE.....

Date:7/12/2024.....

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

| | |
|----------------------------------|--|
| Office Use Only | |
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

| Your Name / Student Number | Course | Date |
|----------------------------|--------|------|
| | | |

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click [here](#).

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

| Tool Name | Brief Description | Link to tool |
|-----------|-------------------|--------------|
| | | |
| | | |

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used.**

| | |
|------------------------------------|---------------------------------|
| [Insert Tool Name] | |
| [Insert Description of use] | |
| [Insert Sample prompt] | [Insert Sample response] |

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Problem 1

I opened the NASA APOD page and entered my email to get API Key. I created a .env file where I saved my API-key and requirements.txt file that has all my dependencies. Then I ensured my .env connection on index.ipynb file by printing my API URL.

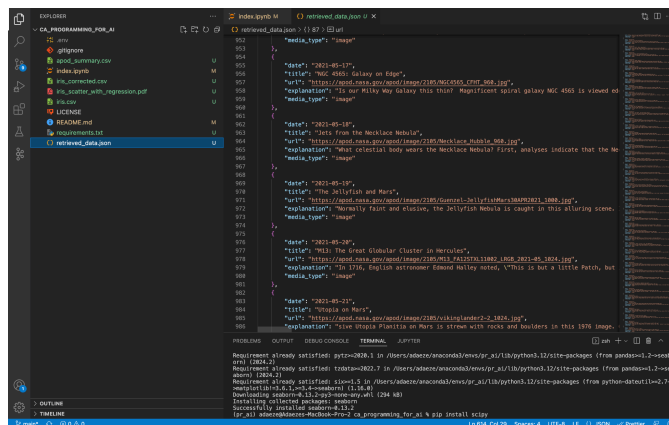
API URL is: <https://api.nasa.gov>

I created a function that gets the following information for 2020-01-01, Date of the picture, title of the picture, URL of the image or video, Explanation of the picture and Media type (image or video). To get this information I had to import REQUEST to enable the GET request.

```
... {'date': '2020-01-01', 'title': 'Betelgeuse Imagined', 'url':  
    'https://apod.nasa.gov/apod/image/2001/BetelgeuseImagined_EsoCalcada_960.jpg', 'explanation': "Why is Betelgeuse fading? No  
    one knows. Betelgeuse, one of the brightest and most recognized stars in the night sky, is only half as bright as it used  
    to be only five months ago. Such variability is likely just normal behavior for this famously variable supergiant, but the  
    recent dimming has rekindled discussion on how long it may be before Betelgeuse does go supernova. Known for its red color,  
    Betelgeuse is one of the few stars to be resolved by modern telescopes, although only barely. The featured artist's  
    illustration imagines how Betelgeuse might look up close. Betelgeuse is thought to have a complex and tumultuous surface  
    that frequently throws impressive flares. Were it to replace the Sun (not recommended), its surface would extend out near  
    the orbit of Jupiter, while gas plumes would bubble out past Neptune. Since Betelgeuse is about 700 light years away, its  
    eventual supernova will not endanger life on Earth even though its brightness may rival that of a full Moon. Astronomers --  
    both amateur and professional -- will surely continue to monitor Betelgeuse as this new decade unfolds. Free  
    Presentation: APOD Editor to show best astronomy images of 2019 -- and the decade -- in NYC on January 3", 'media_type':  
    'image'}
```

I used try and except to handle errors but I could not implement a specific error for invalid or wrong date format.

I generated data from Jan 1st to Dec 31 2020 while adding time.sleep 1 second to respect API rate limit. I used a while loop to iterate through the date range as I did not know the total count of data then tried to append the content of get_apod_data() but could not append JSON data as there is no format for appending JSON data. Retrieved_data.json file was created where collected data is saved and read from.



```
150000  
150001  
150002  
150003  
150004  
150005  
150006  
150007  
150008  
150009  
150010  
150011  
150012  
150013  
150014  
150015  
150016  
150017  
150018  
150019  
150020  
150021  
150022  
150023  
150024  
150025  
150026  
150027  
150028  
150029  
150030  
150031  
150032  
150033  
150034  
150035  
150036  
150037  
150038  
150039  
150040  
150041  
150042  
150043  
150044  
150045  
150046  
150047  
150048  
150049  
150050  
150051  
150052  
150053  
150054  
150055  
150056  
150057  
150058  
150059  
150060  
150061  
150062  
150063  
150064  
150065  
150066  
150067  
150068  
150069  
150070  
150071  
150072  
150073  
150074  
150075  
150076  
150077  
150078  
150079  
150080  
150081  
150082  
150083  
150084  
150085  
150086  
150087  
150088  
150089  
150090  
150091  
150092  
150093  
150094  
150095  
150096  
150097  
150098  
150099  
150100  
150101  
150102  
150103  
150104  
150105  
150106  
150107  
150108  
150109  
150110  
150111  
150112  
150113  
150114  
150115  
150116  
150117  
150118  
150119  
150120  
150121  
150122  
150123  
150124  
150125  
150126  
150127  
150128  
150129  
150130  
150131  
150132  
150133  
150134  
150135  
150136  
150137  
150138  
150139  
150140  
150141  
150142  
150143  
150144  
150145  
150146  
150147  
150148  
150149  
150150  
150151  
150152  
150153  
150154  
150155  
150156  
150157  
150158  
150159  
150160  
150161  
150162  
150163  
150164  
150165  
150166  
150167  
150168  
150169  
150170  
150171  
150172  
150173  
150174  
150175  
150176  
150177  
150178  
150179  
150180  
150181  
150182  
150183  
150184  
150185  
150186  
150187  
150188  
150189  
150190  
150191  
150192  
150193  
150194  
150195  
150196  
150197  
150198  
150199  
150200  
150201  
150202  
150203  
150204  
150205  
150206  
150207  
150208  
150209  
150210  
150211  
150212  
150213  
150214  
150215  
150216  
150217  
150218  
150219  
150220  
150221  
150222  
150223  
150224  
150225  
150226  
150227  
150228  
150229  
150230  
150231  
150232  
150233  
150234  
150235  
150236  
150237  
150238  
150239  
150240  
150241  
150242  
150243  
150244  
150245  
150246  
150247  
150248  
150249  
150250  
150251  
150252  
150253  
150254  
150255  
150256  
150257  
150258  
150259  
150260  
150261  
150262  
150263  
150264  
150265  
150266  
150267  
150268  
150269  
150270  
150271  
150272  
150273  
150274  
150275  
150276  
150277  
150278  
150279  
150280  
150281  
150282  
150283  
150284  
150285  
150286  
150287  
150288  
150289  
150290  
150291  
150292  
150293  
150294  
150295  
150296  
150297  
150298  
150299  
150300  
150301  
150302  
150303  
150304  
150305  
150306  
150307  
150308  
150309  
150310  
150311  
150312  
150313  
150314  
150315  
150316  
150317  
150318  
150319  
150320  
150321  
150322  
150323  
150324  
150325  
150326  
150327  
150328  
150329  
150330  
150331  
150332  
150333  
150334  
150335  
150336  
150337  
150338  
150339  
150340  
150341  
150342  
150343  
150344  
150345  
150346  
150347  
150348  
150349  
150350  
150351  
150352  
150353  
150354  
150355  
150356  
150357  
150358  
150359  
150360  
150361  
150362  
150363  
150364  
150365  
150366  
150367  
150368  
150369  
150370  
150371  
150372  
150373  
150374  
150375  
150376  
150377  
150378  
150379  
150380  
150381  
150382  
150383  
150384  
150385  
150386  
150387  
150388  
150389  
150390  
150391  
150392  
150393  
150394  
150395  
150396  
150397  
150398  
150399  
150400  
150401  
150402  
150403  
150404  
150405  
150406  
150407  
150408  
150409  
150410  
150411  
150412  
150413  
150414  
150415  
150416  
150417  
150418  
150419  
150420  
150421  
150422  
150423  
150424  
150425  
150426  
150427  
150428  
150429  
150430  
150431  
150432  
150433  
150434  
150435  
150436  
150437  
150438  
150439  
150440  
150441  
150442  
150443  
150444  
150445  
150446  
150447  
150448  
150449  
150450  
150451  
150452  
150453  
150454  
150455  
150456  
150457  
150458  
150459  
150460  
150461  
150462  
150463  
150464  
150465  
150466  
150467  
150468  
150469  
150470  
150471  
150472  
150473  
150474  
150475  
150476  
150477  
150478  
150479  
150480  
150481  
150482  
150483  
150484  
150485  
150486  
150487  
150488  
150489  
150490  
150491  
150492  
150493  
150494  
150495  
150496  
150497  
150498  
150499  
150500  
150501  
150502  
150503  
150504  
150505  
150506  
150507  
150508  
150509  
150510  
150511  
150512  
150513  
150514  
150515  
150516  
150517  
150518  
150519  
150520  
150521  
150522  
150523  
150524  
150525  
150526  
150527  
150528  
150529  
150530  
150531  
150532  
150533  
150534  
150535  
150536  
150537  
150538  
150539  
150540  
150541  
150542  
150543  
150544  
150545  
150546  
150547  
150548  
150549  
150550  
150551  
150552  
150553  
150554  
150555  
150556  
150557  
150558  
150559  
150560  
150561  
150562  
150563  
150564  
150565  
150566  
150567  
150568  
150569  
150570  
150571  
150572  
150573  
150574  
150575  
150576  
150577  
150578  
150579  
150580  
150581  
150582  
150583  
150584  
150585  
150586  
150587  
150588  
150589  
150590  
150591  
150592  
150593  
150594  
150595  
150596  
150597  
150598  
150599  
150600  
150601  
150602  
150603  
150604  
150605  
150606  
150607  
150608  
150609  
150610  
150611  
150612  
150613  
150614  
150615  
150616  
150617  
150618  
150619  
150620  
150621  
150622  
150623  
150624  
150625  
150626  
150627  
150628  
150629  
150630  
150631  
150632  
150633  
150634  
150635  
150636  
150637  
150638  
150639  
150640  
150641  
150642  
150643  
150644  
150645  
150646  
150647  
150648  
150649  
150650  
150651  
150652  
150653  
150654  
150655  
150656  
150657  
150658  
150659  
150660  
150661  
150662  
150663  
150664  
150665  
150666  
150667  
150668  
150669  
150670  
150671  
150672  
150673  
150674  
150675  
150676  
150677  
150678  
150679  
150680  
150681  
150682  
150683  
150684  
150685  
150686  
150687  
150688  
150689  
150690  
150691  
150692  
150693  
150694  
150695  
150696  
150697  
150698  
150699  
150700  
150701  
150702  
150703  
150704  
150705  
150706  
150707  
150708  
150709  
150710  
150711  
150712  
150713  
150714  
150715  
150716  
150717  
150718  
150719  
150720  
150721  
150722  
150723  
150724  
150725  
150726  
150727  
150728  
150729  
150730  
150731  
150732  
150733  
150734  
150735  
150736  
150737  
150738  
150739  
150740  
150741  
150742  
150743  
150744  
150745  
150746  
150747  
150748  
150749  
150750  
150751  
150752  
150753  
150754  
150755  
150756  
150757  
150758  
150759  
150760  
150761  
150762  
150763  
150764  
150765  
150766  
150767  
150768  
150769  
150770  
150771  
150772  
150773  
150774  
150775  
150776  
150777  
150778  
150779  
150780  
150781  
150782  
150783  
150784  
150785  
150786  
150787  
150788  
150789  
150790  
150791  
150792  
150793  
150794  
150795  
150796  
150797  
150798  
150799  
150800  
150801  
150802  
150803  
150804  
150805  
150806  
150807  
150808  
150809  
150810  
150811  
150812  
150813  
150814  
150815  
150816  
150817  
150818  
150819  
150820  
150821  
150822  
150823  
150824  
150825  
150826  
150827  
150828  
150829  
150830  
150831  
150832  
150833  
150834  
150835  
150836  
150837  
150838  
150839  
150840  
150841  
150842  
150843  
150844  
150845  
150846  
150847  
150848  
150849  
150850  
150851  
150852  
150853  
150854  
150855  
150856  
150857  
150858  
150859  
150860  
150861  
150862  
150863  
150864  
150865  
150866  
150867  
150868  
150869  
150870  
150871  
150872  
150873  
150874  
150875  
150876  
150877  
150878  
150879  
150880  
150881  
150882  
150883  
150884  
150885  
150886  
150887  
150888  
150889  
150890  
150891  
150892  
150893  
150894  
150895  
150896  
150897  
150898  
150899  
150900  
150901  
150902  
150903  
150904  
150905  
150906  
150907  
150908  
150909  
150910  
150911  
150912  
150913  
150914  
150915  
150916  
150917  
150918  
150919  
150920  
150921  
150922  
150923  
150924  
150925  
150926  
150927  
150928  
150929  
150930  
150931  
150932  
150933  
150934  
150935  
150936  
150937  
150938  
150939  
150940  
150941  
150942  
150943  
150944  
150945  
150946  
150947  
150948  
150949  
150950  
150951  
150952  
150953  
150954  
150955  
150956  
150957  
150958  
150959  
150960  
150961  
150962  
150963  
150964  
150965  
150966  
150967  
150968  
150969  
150970  
150971  
150972  
150973  
150974  
150975  
150976  
150977  
150978  
150979  
150980  
150981  
150982  
150983  
150984  
150985  
150986  
150987  
150988  
150989  
150990  
150991  
150992  
150993  
150994  
150995  
150996  
150997  
150998  
150999  
151000  
151001  
151002  
151003  
151004  
151005  
151006  
151007  
151008  
151009  
151010  
151011  
151012  
151013  
151014  
151015  
151016  
151017  
151018  
151019  
151020  
151021  
151022  
151023  
151024  
151025  
151026  
151027  
151028  
151029  
151030  
151031  
151032  
151033  
151034  
151035  
151036  
151037  
151038  
151039  
151040  
151041  
151042  
151043  
151044  
151045  
151046  
151047  
151048  
151049  
151050  
151051  
151052  
151053  
151054  
151055  
151056  
151057  
151058  
151059  
151060  
151061  
151062  
151063  
151064  
151065  
151066  
151067  
151068  
151069  
151070  
151071  
151072  
151073  
151074  
151075  
151076  
151077  
151078  
151079  
151080  
151081  
151082  
151083  
151084  
151085  
151086  
151087  
151088  
151089  
151090  
151091  
151092  
151093  
151094  
151095  
151096  
151097  
151098  
151099  
151100  
151101  
151102  
151103  
151104  
151105  
151106  
151107  
151108  
151109  
151110  
151111  
151112  
151113  
151114  
151115  
151116  
151117  
151118  
151119  
151120  
151121  
151122  
151123  
151124  
151125  
151126  
151127  
151128  
151129  
151130  
151131  
151132  
151133  
151134  
151135  
151136  
151137  
151138  
151139  
151140  
151141  
151142  
151143  
151144  
151145  
151146  
151147  
151148  
151149  
151150  
151151  
151152  
151153  
151154  
151155  
151156  
151157  
151158  
151159  
151160  
151161  
151162  
151163  
151164  
151165  
151166  
151167  
151168  
151169  
151170  
151171  
151172  
151173  
151174  
151175  
151176  
151177  
151178  
151179  
151180  
151181  
151182  
151183  
151184  
151185  
151186  
151187  
151188  
151189  
151190  
151191  
151192  
151193  
151194  
151195  
151196  
151197  
151198  
151199  
151200  
151201  
151202  
151203  
151204  
151205  
151206  
151207  
151208  
151209  
151210  
151211  
151212  
151213  
151214  
151215  
151216  
151217  
151218  
151219  
151220  
151221  
151222  
151223  
151224  
151225  
151226  
151227  
151228  
151229  
151230  
151231  
151232  
151233  
151234  
151235  
151236  
151237  
151238  
151239  
151240  
151241  
151242  
151243  
151244  
151245  
151246  
151247  
151248  
151249  
151250  
151251  
151252  
151253  
151254  
151255  
151256  
151257  
151258  
151259  
151260  
151261  
151262  
151263  
151264  
151265  
151266  
151267  
151268  
151269  
151270  
151271  
151272  
151273  
151274  
151275  
151276  
151277  
151278  
151279  
151280  
151281  
151282  
151283  
151284  
151285  
151286  
151287  
151288  
151289  
151290  
151291  
151292  
151293  
151294  
151295  
151296  
151297  
151298  
151299  
151300  
151301  
151302  
151303  
151304  
151305  
151306  
151307  
151308  
151309  
151310  
151311  
151312  
151313  
151314  
151315  
151316  
151317  
151318  
151319  
151320  
151321  
151322  
151323  
151324  
151325  
151326  
151327  
151328  
151329  
151330  
151331  
151332  
151333  
151334  
151335  
151336  
151337  
151338  
151339  
151340  
151341  
151342  
151343  
151344  
151345  
151346  
151347  
151348  
151349  
151350  
151351  
151352  
151353  
151354  
151355  
151356  
151357  
151358  
151359  
151360  
151361  
151362  
151363  
151364  
151365  
151366  
151367  
151368  
151369  
151370  
151371  
151372  
151373  
151374  
151375  
151376  
151377  
151378  
151379  
151380  
151381  
151382  
151383  
151384  
151385  
151386  
151387  
151388  
151389  
151390  
151391  
151392  
151393  
151394  
151395  
151396  
151397  
151398  
151399  
151400  
151401  
151402  
151403  
151404  
151405  
151406  
151407  
151408  
151409  
151410  
151411  
151412  
151413  
151414  
151415  
151416  
151417  
151418  
151419  
151420  
151421
```

Problem 2

I wrote a function that reads the apod_data.json file and saves the data in a Python dictionary. I used try and except to handle errors such as file not found and JSON errors.

```
Error: 'apod_data.json' file not found.
```

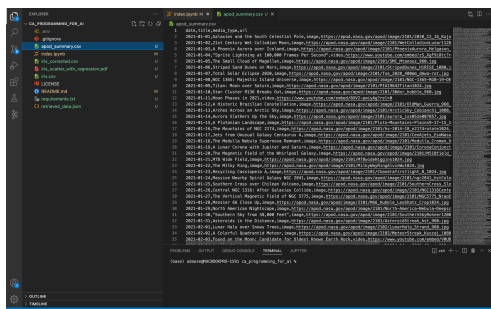
The Read_apod_data function returns only the date and title of each entry

```
{'date': '2021-01-01', 'title': 'Galaxies and the South Celestial Pole'}
{'date': '2021-01-02', 'title': '21st Century Wet Collodion Moon'}
{'date': '2021-01-03', 'title': 'A Phoenix Aurora over Iceland'}
{'date': '2021-01-04', 'title': 'Sprite Lightning at 100,000 Frames Per Second'}
{'date': '2021-01-05', 'title': 'The Small Cloud of Magellan'}
{'date': '2021-01-06', 'title': 'Striped Sand Dunes on Mars'}
{'date': '2021-01-07', 'title': 'Total Solar Eclipse 2020'}
{'date': '2021-01-08', 'title': 'NGC 1365: Majestic Island Universe'}
{'date': '2021-01-09', 'title': 'Titan: Moon over Saturn'}
{'date': '2021-01-10', 'title': 'Star Cluster R136 Breaks Out'}
{'date': '2021-01-11', 'title': 'Moon Phases in 2021'}
{'date': '2021-01-12', 'title': 'A Historic Brazilian Constellation'}
{'date': '2021-01-13', 'title': 'Arches Across an Arctic Sky'}
{'date': '2021-01-14', 'title': 'Aurora Slathers Up the Sky'}
{'date': '2021-01-15', 'title': 'A Plutonian Landscape'}
{'date': '2021-01-16', 'title': 'The Mountains of NGC 2174'}
```

To get the total count for images, videos and longest text, I created a new function, initialized image and video count to 0 then looped through the data checking for where media type is either image or video then increasing their count. For the longest text I used Python library max() and len() to get the entry with the largest date.

```
Total count of Images: 328
Total count of Videos: 37
Entry with the longest explanation:
{
  "date": "2021-04-16",
  "title": "The Doubly Warped World of Binary Black Holes",
  "url": "https://www.youtube.com/embed/rQcKIN9vj3U?rel=0",
  "explanation": "Light rays from accretion disks around a pair of orbiting supermassive black holes make their way through the warped space-time produced by extreme gravity in this stunning computer visualization. The simulated accretion disks have been given different false color schemes, red for the disk surrounding a 200-million-solar-mass black hole, and blue for the disk surrounding a 100-million-solar-mass black hole. That makes it easier to track the light sources, but the choice also reflects reality. Hotter gas gives off light closer to the blue end of the spectrum and material orbiting smaller black holes experiences stronger gravitational effects that produce higher temperatures. For these masses, both accretion disks would actually emit most of their light in the ultraviolet though. In the video, distorted secondary images of the blue black hole, which show the red black hole's view of its partner, can be found within the tangled skein of the red disk warped by the gravity of the blue black hole in the foreground. Because we're seeing red's view of blue while also seeing blue directly, the images allow us to see both sides of blue at the same time. Red and blue light originating from both black holes can be seen in the innermost ring of light, called the photon ring, near their event horizons. Astronomers expect that in the not-too-distant future they'll be able to detect gravitational waves, ripples in space-time, produced when two supermassive black holes in a system much like the one simulated here spiral together and merge.",
  "media_type": "video"
}
```

I used a function to extract and write data to apod_summary CSV. The CSV file contains appropriate column headers. New data is appended to existing data, see output below



Problem 3

I imported numpy and then created 20 rows and 5 columns with .random() ranging from 10 to 100, which I named data.

```
array([[13, 85, 52, 47, 98],
       [85, 62, 91, 79, 81],
       [85, 41, 63, 55, 45],
       [39, 49, 90, 21, 63],
       [95, 84, 78, 18, 84],
       [57, 90, 29, 64, 35],
       [24, 96, 39, 45, 40],
       [37, 75, 92, 73, 11],
       [83, 15, 98, 21, 29],
       [26, 54, 99, 95, 92],
       [32, 91, 65, 37, 21],
       [12, 44, 67, 36, 72],
       [76, 67, 40, 29, 59],
       [85, 52, 49, 63, 26],
       [46, 71, 58, 29, 16],
       [16, 66, 89, 23, 61],
       [80, 66, 29, 89, 94],
       [52, 12, 43, 12, 13],
       [61, 53, 29, 56, 32],
       [94, 71, 58, 62, 35]])
```

```
# Sums each row to check if they are even or odd.
np.sum(data, axis=1)
```

Summed all data to check if they were odd or even, since they were not all even I had to loop through each row, rows that summed up to even were displayed and for the odd rows, I subtracted 1 from the first data on that row to make it even. See output below

```
260
192
270
334
276
236
186
194
204
228
306
332
304
162
308
270
244
306
252
216
```

Multiples of 5 are numbers when divided by 5 has 0 remainder, so to check if the total is a multiple of 5 I had to sum the whole array and divide it by 5, if it returns 0 it is a multiple of 5 and the total is displayed but if it is not the remainder is subtracted from one element in the row to make it a multiple of 5.

For this, an AND operator is used to ensure the values displayed are divisible by both 3 and 5.

The mean of the array is gotten, numpy has a function for this. Then a check is made using an if statement to check for numbers greater than 75 and the numbers found were replaced by mean. Using numpy functions I calculated the mean, Standard Deviation, median and Variance for each column

```
Mean of the array: 46.36
Standard Deviation of the array: 17.010890629241022
Median of the array: 55.0
Variance for each column: [314.84 256.3275 129.11 314.86
```

Problem 4

I used iris data from Moodle a copy has been included in the project and all necessary libraries have been imported. There are 150 rows and 5 columns in the dataset. There are 4 numerical columns and a Nominal Categorical Feature (Species), see below for the names of the columns

```
Index(['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width',  
      'Species'],
```

There are 3 species of flower

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

The 2 rows with incorrect data were fixed

```
# Checking fix.  
updatedRows = irisData.iloc[[35,38]]  
print(updatedRows)
```

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|----|--------------|-------------|--------------|-------------|---------|
| 35 | 4.9 | 3.1 | 1.5 | 0.2 | setosa |
| 38 | 4.9 | 3.6 | 1.4 | 0.1 | setosa |

Two extra columns were added and they are derived from the ratio of the width and length of existing features

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species | Petal.Ratio | Sepal.Ratio |
|---|--------------|-------------|--------------|-------------|---------|-------------|-------------|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | setosa | 7.000000 | 1.457143 |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | setosa | 7.000000 | 1.633333 |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | setosa | 6.500000 | 1.468750 |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | setosa | 7.500000 | 1.483871 |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | setosa | 7.000000 | 1.388889 |

This new dataset is saved to CSV

4.4 The two pairs of features that have the highest positive correlation are **petal width and petal length with 0.962786 as they are close to 1** and the highest negative correlation are **sepal length and width with -0.119003** this indicates a weak negative linear

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Petal.Ratio | Sepal.Ratio |
|--------------|--------------|-------------|--------------|-------------|-------------|-------------|
| Sepal.Length | 1.000000 | -0.119003 | 0.871693 | 0.817809 | -0.564533 | 0.728583 |
| Sepal.Width | -0.119003 | 1.000000 | -0.429425 | -0.365543 | 0.335022 | -0.750309 |
| Petal.Length | 0.871693 | -0.429425 | 1.000000 | 0.962786 | -0.684523 | 0.836973 |
| Petal.Width | 0.817809 | -0.365543 | 0.962786 | 1.000000 | -0.733923 | 0.752458 |
| Petal.Ratio | -0.564533 | 0.335022 | -0.684523 | -0.733923 | 1.000000 | -0.589552 |
| Sepal.Ratio | 0.728583 | -0.750309 | 0.836973 | 0.752458 | -0.589552 | 1.000000 |

From these statistics, I can infer that petal width and length are highly correlated and have a strong linear relationship meaning as petal length increases, petal width increases as well. While sepal width and length are weakly related meaning a change in sepal width might not lead to a change in sepal length.