Test 2 - Test Log

Description of test	Test data to be used (if required)	Expected outcome	Actual outcome	Comments and intended actions
Running the program to find syntax errors	**N/A**	**An error message will occur**	Please enter a name for the booking: dsfgfd Please enter your email address: fdg The email entered did not match expected form Please enter your email address:	**Code can begin without any syntax errors. Will test if correct data can be inputted**
Attempting to input name and email in correct format	**Intended Data**	**The program should capture my inputs and move me on to the next part of the program**	Please enter a name for the booking: William Hemingway Please enter your email address: WillHeming@mail.com Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY):	**Program can capture the correct format.**

Continuing inputting the correct data to see if any errors are printed to terminal	**Intended data**	**The program should continue through each sub routine**	`user_date = input('Please enter date you wish to visit`
			`try:`
			`datetime.strptime(user_dat, "%d/%m/%Y").date()`
			`except:`
			`print("Sorry, you did not enter a valid date")`
			`flag = True`
			
			Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/05/2024
			Sorry, you did not enter a valid date
			Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20052024
			Sorry, you did not enter a valid date
				Program does not perform the correct validation check to capture the date.
				
				The variable is misnamed within the validation check. Will correct to 'user_date' and re run the test

Attempting to input the correct format of date to see if it can capture it correctly, after correcting the variable name.
`user_date = input('Please enter date you wish to vi`
`try:`
`datetime.strptime(user_date, "%d/%m/%Y").date()`
`except:`
`print("Sorry, you did not enter a valid date")`
`flag = True`
	Intended data	**Program should capture the data when inputted correctly**	Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/05/2024
			Please enter the number of adult tickets required: yay!8
				Program successfully continues whilst capturing the correct data

<table>
<tr>
<td>Will continue inputting valid data until an error occurs</td>
<td>Intended data</td>
<td>Program should continue collecting data from me before outputting it</td>
<td>

```
input('Please enter the number of {} tickets required: '.format(ticket_type))
```

```
Please enter the number of adult tickets required: 5
Please enter the number of junior tickets required: 4
Traceback (most recent call last):
  File "H:\Task2code LL-000018217-hemingway-w.py", line 132, in <module>
    not_junior = check_age(junior_tickets)
  File "H:\Task2code LL-000018217-hemingway-w.py", line 60, in check age
    for i in range(num_junior):
TypeError: 'str' object cannot be interpreted as an integer
```

</td>
<td>An error has occurred when using the input for junior tickets as a parameter for the loop. This is likely caused due to not capturing the input as an integer.

When the ticket amount gets captured, it gets formatted. This will make it no longer a integer as the ticket_type variable is a string. To solve this issue I will print what the variable looks like after the formatting to see what needs to be dropped in order for it to be used as an integer again</td>
</tr>
<tr>
<td>Running the program to find out what is stored after the variable has been formatted

```
def get_ticket_numbers (ticket_type):

    flag = True

    while True:
        num_tickets = input('Please er

        try:
            int(num_tickets)
        except:
            print("Sorry, you did not
            flag = True
        else:
            print(num_tickets)
            return num_tickets
```

</td>
<td>Intended data</td>
<td>Variable should have the number I inputted as well as additional string characters</td>
<td>

```
Please enter the number of adult tickets required: 5
5
Please enter the number of junior tickets required: 23
23
```

</td>
<td>I have realized the formatting function is used to help output the message and does not change what is captured. I will turn my attention and instead I will look at how the list is created to see if any issues are within it’s syntax.

```
for i in range(num_junior):
    flag = True
```

When used for the loop, it is not specified as an int. Will attempt to run the program again whilst forcing the variable to be an int().</td>
</tr>
<tr>
<td>Running the program after forcing the variable in the for loop parameters to be a integer.

```
for i in range(int(num_junior)):
    flag = True
```

</td>
<td>Intended data</td>
<td>Should allow me to go through the list and input an age for all the junior tickets</td>
<td>

```
Please enter the number of adult tickets required: 4
Please enter the number of junior tickets required: 4
Please enter the age for junior ticket 0 : 5
Please enter the age for junior ticket 1 : 8
Please enter the age for junior ticket 2 : 54
Sorry you have entered an age over 16. This ticket has been chnaged to an adult t
icket
Please enter the age for junior ticket 3 : 7
```

</td>
<td>Program will now loop depending on how many tickets you inputted. In this print screen however, it outputs ‘junior ticket 0’ first. Will change this so it outputs the number + 1 to make the output make sense.</td>
</tr>
<tr>
<td>Re-running to see if the list outputs the now correct iterations of the tickets.

Before:

```
input('Please enter the age for junior ticket {} : '.format(i))
```

After:

```
age for junior ticket {} : '.format(i+1))
```

</td>
<td>Intended data</td>
<td>List will output starting with 1 instead of 0</td>
<td>

```
Please enter the number of adult tickets required: 3
Please enter the number of junior tickets required: 5
Please enter the age for junior ticket 1 : 6
Please enter the age for junior ticket 2 : 5
Please enter the age for junior ticket 3 : 7
Please enter the age for junior ticket 4 : 4
Please enter the age for junior ticket 5 : 8
```

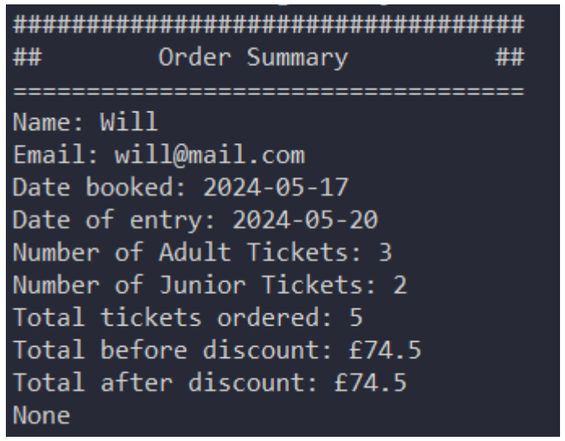
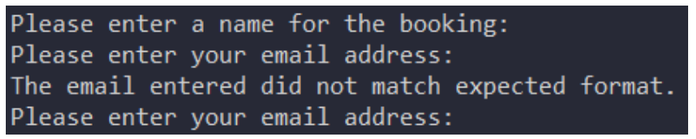
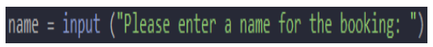
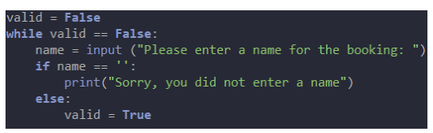
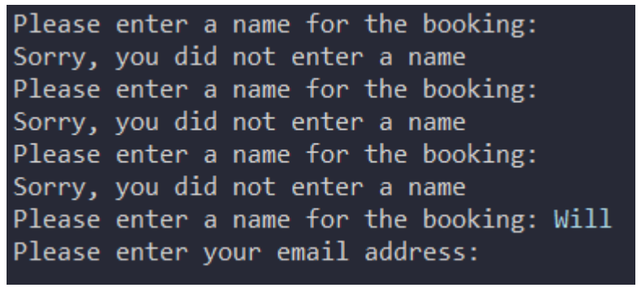
</td>
<td>Program outputs correct list format now</td>
</tr>
</table>

Will run program will all child tickets of the correct age to see if it causes any 0 value errors with the check_age function	**Intended data**	**Program should be able to capture the 0 value and discard of it without printing an error**	Please enter the number of junior tickets required: 3 Please enter the age for junior ticket 1 : 6 Please enter the age for junior ticket 2 : 7 Please enter the age for junior ticket 3 : 6 Traceback (most recent call last): File "H:\Task2code LL-000018217-hemingway-w.py", line 135, in <module> final_adult_tickets = adult_tickets + not_junior TypeError: can only concatenate str (not "int") to str 135 final_adult_tickets = adult_tickets + not_junior 136 final_junior_tickets = junior_tickets - not_junior	**Program breaks if no junior tickets are over age. Will check if this is due to a 0 error or another error within the programming**
Testing the same subroutine however, this time there will be child tickets over age to see if the program can capture them and continue	**Intended data**	**Program should tally the tickets over 16 and store them to be used on line 135**	Please enter the number of junior tickets required: 5 Please enter the age for junior ticket 1 : 5 Please enter the age for junior ticket 2 : 89 Sorry you have entered an age over 16. This ticket has been chnaged to an adult t icket Please enter the age for junior ticket 3 : 17 Sorry you have entered an age over 16. This ticket has been chnaged to an adult t icket Please enter the age for junior ticket 4 : 16 Sorry you have entered an age over 16. This ticket has been chnaged to an adult t icket Please enter the age for junior ticket 5 : 3 Traceback (most recent call last): File "H:\Task2code LL-000018217-hemingway-w.py", line 135, in <module> final_adult_tickets = adult_tickets + not_junior TypeError: can only concatenate str (not "int") to str **Produced the same error**	**Program breaks no matter what the tally on the variable is. This is due to how it is being used on line 135. Will attempt to force the variables into int() to concatenate them**
Changed the calculations on line 135,136 to force an integer to allow for the functions to work correctly 135 final_adult_tickets = int(adult_tickets) + int(not_junior) 136 final_junior_tickets = int(junior_tickets) - int(not_junior) 137	**Intended data**	**Program should finalize taking inputs and finally produce a summary of my order.**	**(Changed to dark theme as the program was starting to hurt my eyes. This is still the same step in the program)** ## ## Order Summary ## == Name: William Email: will@mail.com Date booked: 2024-05-17 Date of entry: 2024-05-20 Number of Adult Tickets: 6 Number of Junior Tickets: 6 Total tickets ordered: 9 Toatal before discount: £119.0 Total after discount: £-1666.0 None Please enter the number of adult tickets required: 5 Please enter the number of junior tickets required: 4 Please enter the age for junior ticket 1 : 4 Please enter the age for junior ticket 2 : 90 Sorry you have entered an age over 16. This ticket has been chnaged to an adult ticket Please enter the age for junior ticket 3 : 2 Please enter the age for junior ticket 4 : 4 ("Number of Adult Tickets: {}".format(final_total.get("num_adults"))) ("Number of Junior Tickets: {}".format(final_total.get("num_adults")))	**The order summary does not match up with the intended data. It correctly outputs the number of adult tickets (5 + 1 child ticket over 16) but then doesn't correctly output the child ticket.** **The prices are also unrealistic but I will solve those issues in later testing** **When outputting, the program is outputting the same variable for both adult and ticket prices. Will change the formatted variable in order for the program to output correct amounts.**

Changed the outputting block so that Junior tickets is displaying the correct amount. Running program to test it.
 	
`format(final_total.get("num_adults")))`
`.format(final_total.get("num_junior")))`
	Intended data	**Program should now output the correct number of junior tickets**	`####################################`
			`##  Order Summary  ##`
			`====================================`
			`Name: Will`
			`Email: will@mail.com`
			`Date booked: 2024-05-17`
			`Date of entry: 2024-05-20`
			`Number of Adult Tickets: 6`
			`Number of Junior Tickets: 3`
			`Total tickets ordered: 9`
			`Toatal before discount: £119.0`
			`Total after discount: £-1666.0`
			`None`
			Entered same data as on previous test
				Program now outputs the correct amount of both adult and junior tickets.
				
				The cost is still not correct. Will be looking into that area of the program next before I starting extreme and erroneous testing.
				
				`def calculate_subtotal (adults,juniors):`
				`adult_subtotal = adults * 17.50`
				`junior_subtotal = juniors + 11.00`
				
				When calculating price. Juniors charging £11 per child but instead adding the amount of children to 11. Changing calculation to 'juniors * 11.00'

After fixing the calculations of child tickets, re running to test what has changed in the final summary. Using the same data as the previous 2 tests.
 	
`def calculate_subtotal (adults,juniors):`
`adult_subtotal = adults * 17.50`
`junior_subtotal = juniors * 11.00 # Ch`
	Intended data	**The calculations will still be incorrect due to the total after discount being a negative, but the total before discount should output 138.**	`####################################`
			`##  Order Summary  ##`
			`====================================`
			`Name: Will`
			`Email: will@mail.com`
			`Date booked: 2024-05-17`
			`Date of entry: 2024-05-20`
			`Number of Adult Tickets: 6`
			`Number of Junior Tickets: 3`
			`Total tickets ordered: 9`
			`Total before discount: £138.0`
			`Total after discount: £-1932.0`
			`None`
				The total before discount is now the correct amount due to:
				
				6*17.50 = 105
				3*11.00 = 33
				
				105 + 33 = £138
				
				Will now begin to look at the discount calculations to find what is going wrong
				
				When totally the discounted price, it is subtracting the (total * 15) from the original price. Meaning it will always be 14 times the price, in the negatives. I will solve the equation to ensure it is a 15% discount applied instead.

Changed the number in the variable to be 015 instead of 15 to act as 15%. This should now apply the correct discount code. Running with same data as previous test to ensure.
 	
`if total_tickets >= 8:`
`discount = 0.15 # Was originally 15.`
`else:`
`discount = 0`
	Intended data	**Should output the correct price after discount. Likely could not be to 2 decimal places like it should be due to currency**	`####################################`
			`##  Order Summary  ##`
			`====================================`
			`Name: Will`
			`Email: will@mail.com`
			`Date booked: 2024-05-17`
			`Date of entry: 2024-05-20`
			`Number of Adult Tickets: 6`
			`Number of Junior Tickets: 3`
			`Total tickets ordered: 9`
			`Total before discount: £138.0`
			`Total after discount: £117.3`
			`None`
				15% of 138 = 20.7
				138 – 20.7 = 117.3
				
				The discount is applied correctly when the group is over 8 people or larger.
				
				Will check calculation is still fine with groups that do not get the discount.

Running test with a group smaller than 8 to see if the discount will not apply	**Intended data (3 adults 2 children)**	**Program should display the same number in both the price markups within the summary**		**Program now functions as intended when inputting data within the parameters of the code. Will now commence extreme data testing to see if the program is capable of processing or removing extreme, null and erroneous values.**
Running the program putting null data whenever prompted to input.	**Null data (nothing)**	**Program should capture all attempts at putting null data and loop until I put the a valid input in.**		**The program allowed me to enter nothing as my name. Will put in place a check to ensure at least a character is entered**
Changed the program to check for null data inputs  	**Null data**	**Program will now loop until I enter a name**		**Program will not continue until a name is inputted.** **Will next test the email**

<table>
<tr>
<td>Testing to see if the email will pass through even if the characters that are being checked are used, but in the wrong place.</td>
<td>Erroneous data</td>
<td>Program will allow me to input a invalid email</td>
<td>

```
Please enter a name for the booking: Will
Please enter your email address: name.com@mail
Please enter date you wish to visit Recoats Adventure Park
```

```
match = False

while match == False: # My loop that will be used to match the 2 inputted em
    while flag:
        email = input ('Please enter your email address: ')
        if "@" not in email or not "." in email:
            print("The email entered did not match expected format. Please t
            flag = True
        else:
            flag = False

    #Additional validation check added by me to ensure the 2 emails match
    email2 = input ('Please re-enter to confirm your email address:  ')
    if email != email2:
        print("Emails do not match. Please try again.")
        flag = True
    else:
        match = True

return email
```

</td>
<td>Program allows me to enter a incorrect email. Will print out the email to the user and as them to input it again to ensure it is the correct email. (As shown in the second image)</td>
</tr>
<tr>
<td>Testing to see if my validation check will loop until 2 matching emails are inputted.</td>
<td>Erroneous data</td>
<td>Program will loop until 2 matching emails are inputted</td>
<td>

```
Please enter a name for the booking: Will
Please enter your email address: will@mail.com
Please re-enter to confirm your email address:  notwill@mail.com
Emails do not match. Please try again.
Please enter your email address: will@mail.com
Please re-enter to confirm your email address:  will@mail.com
Please enter date you wish to visit Recoats Adventure Park (DD/MM/Y
```

</td>
<td>Program now validates that the email is what the user intended to input. Ensuring it is their email.</td>
</tr>
<tr>
<td>Testing the visit date input by inputting a date 300 years into the future</td>
<td>Extreme data</td>
<td>Program will allow me to input it as there is no set date within the task brief or programmed into the code</td>
<td>

```
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/05/2324
Please enter the number of adult tickets required:
```

</td>
<td>Program accepts the inputted
Will now check incorrect formatting</td>
</tr>
<tr>
<td>Testing to see if the date will accept an incorrect format</td>
<td>Erroneous data</td>
<td>Program will loop until a correct date format is submitted</td>
<td>

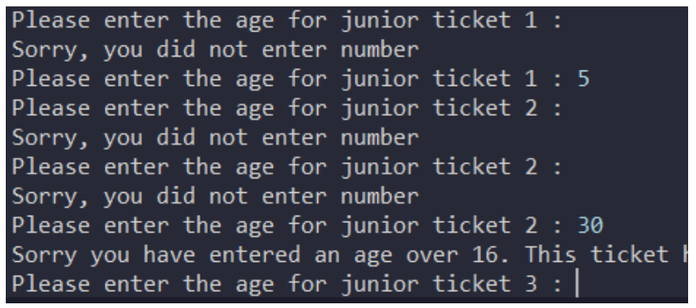
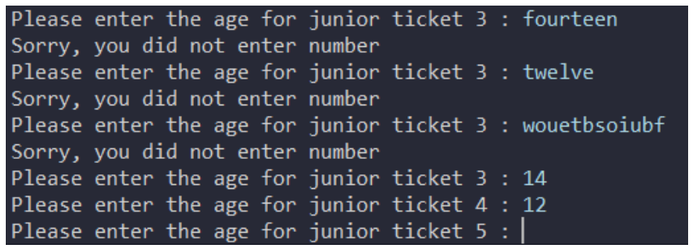
```
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 45/607/33
Sorry, you did not enter a valid date
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 45
Sorry, you did not enter a valid date
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 45/12/2024
Sorry, you did not enter a valid date
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/
Sorry, you did not enter a valid date
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/25/2024
Sorry, you did not enter a valid date
Please enter date you wish to visit Recoats Adventure Park (DD/MM/YYYY): 20/12/2012
Please enter the number of adult tickets required:
```

</td>
<td>Date input is strong in terms of robustness however, it allows you to input a date previous to the current date.

I am not trained within the datetime library so am unable to fix it, but this is a logic error that needs to be appended.</td>
</tr>
<tr>
<td>Testing to see how robust the ticket number inputting routine works</td>
<td>Extreme data</td>
<td>Program should allow any number (no matter how big) to be inputted as the task information does not limit how many tickets can be booked at once</td>
<td>

```
Please enter the number of adult tickets required: 9999999
Please enter the number of junior tickets required: 999999
Please enter the age for junior ticket 1 :
```

</td>
<td>There should be a limit on this however, the task information does not specify and therefore no limit has been set.</td>
</tr>
</table>

Testing same subroutine to see if it can catch null data entries	**Null data**	**Program should loop until I enter an age**		**Program successfully loops until a age is entered**
Testing same subroutine to see what happens if I enter a string for their age	**Erroneous data**	**Program should loop saying I should enter a number**		**Program prints same error message as if it was a null data entry. This is what should happen.**

Add more rows and tables as required

<u>All test are complete. Code runs as functions and has built in robustness in case of user error</u>