

Pizzaz cinema

Due date 11:59PM Sunday 11 April 2021 (local Sydney time)

Be aware of daylight savings changes on 4 April 2021.

Changes to description

28 March 2021

- seat number for one person is now #17 (see --book)
- seat allocation formula changed from $(2N+1)$ -> $(2N-1)$ (see --group)
- room size and room capacity are now differentiated, described and examples updated (see --group)
- added note about what to do when questions with fixed responses are not satisfied (see Tips and error cases to consider)

A message to all students about posting on Ed

This is an assignment and staff are not permitted to give guidance on your code, or how to solve the specific problem. That is the purpose of the assessment that you are required to perform to achieve the grade.

You may ask clarification questions about the assignment description. This is often necessary to implement functionality that is otherwise ambiguous.

The assignment description is not intended to be complete and you can confirm your assumptions in a form of a question. In asking the question you should be quoting the description you are asking about.

If you have a question to ask on Ed please search before asking. However, remember that **you should not be posting any assignment code publicly, as this would constitute academic dishonesty.**

Also, do not wait too long before starting. This assignment needs time and sustained effort.

Problem description

You are writing an interactive program for a local cinema. The program will allow users to view the available movies, order tickets for one or multiple persons, and make transactions on the orders.

When ordering tickets, further information about each person can be asked. The order can also determine if the number of persons in a group can see the movie. Finally, the purchase will involve a transaction between the cinema involving calculations of finance.

Program data

The program has a predefined schedule, room numbers and capacities. For the purposes of this assignment, this information is fixed and will not change. You can include this data in your program source code.

Schedule of movies, their times and room

The Shining. 1980. 2h 26m. 10:00. Room 1

Your Name. 2016. 1h 52m. 13:00. Room 1

Fate/Stay Night: Heaven's Feel - III. Spring Song. 2020. 2h 0m. 15:00. Room 1

The Night Is Short, Walk on Girl. 2017. 1h 32m. 17:30. Room 1

The Truman Show. 1998. 1h 47m. 19:30. Room 1

Genocidal Organ. 2017. 1hr 55m. 21:45. Room 1

Jacob's Ladder. 1990. 1h 56m. 10:00. Room 2

Parasite. 2019. 2h 12m. 12:15. Room 2

The Dark Knight. 2008. 2h 32min. 14:45. Room 2

Blade Runner 2049. 2017. 2h 44m. 17:45. Room 2

The Mist. 2007. 2h 6m. 21:00. Room 2

Demon Slayer: Mugen Train. 2020. 1h59min. 23:20. Room 2

The Matrix. 1999. 2h 16m. 10:00. Room 3

Inception. 2010. 2h 42m. 11:30. Room 3

Shutter Island. 2010. 2h 19m. 14:30. Room 3

Soul. 2020. 1hr 40m. 17:00. Room 3

Mrs. Brown. 1997. 1h 41min. 19:00. Room 3

Peppa Pig: Festival of Fun. 2019. 1h 8min. 21:00. Room 3

Titanic. 1997. 3h 30min. 22:15. Room 3

Room information

Each room has a different capacity and distinct areas for seating. The total number of seats will become important when considering group bookings.

Key

* - represents empty space for walkways

A,B,C,D,E, - represent a section of seating

Room 1 - 35 seats –

```
*****
*CCC*AAAA*BBB*DD
*CCC*AAAA*BBB*DD
*C*C*AAAA*BBB*DD
*****
```

Room 2 - 136 seats-

```
*****
AAA**BBBBB**CCCCC**EEE
AAA**BBBBB**CCCCC**EEE
AAA**BBBBB**CCCCC**EEE
AAA**BBBBB**CCCCC**EEE
AAA**BBBBB**CCCCC**EEE
AAA*****EEE
AAAA*DDDDDDDD*EEEE
AAAA*DDDDDDDD*EEEE
AAAA*DDDDDDDD*EEEE
*****
```

Room 3 - 42 seats

```
AAA*BBBB
AAA*BBBB
AAA*BBBB
AAA*BBBB
AAA*BBBB
AAA*BBBB
```

Program operation

The program will be invoked by running the command

Sorry. This program does not recognise the switch options.

Bye.

```
$ python3 pizzaz.py haberjabby --book
```

```
=====
~ Welcome to Pizzaz cinema ~
=====
```

Sorry. This program does not recognise the switch options.

Bye.

```
$ python3 pizzaz.py --group abc
```

```
=====
~ Welcome to Pizzaz cinema ~
=====
```

Sorry. This program does not recognise the switch options.

Bye.

Switch **--show**

To complete this, you would need to have completed week 3 lecture and tutorial

With this switch your program will show all the movies starting from the time provided. The parameter `<time now>` is a 24 hour time format. It will affect the output.

Suppose the user wished to see all movies starting from 19:00. The program would print the following:

```
$ python3 pizzaz.py --show 19:00
```

```
=====
~ Welcome to Pizzaz cinema ~
```

```
~~~~~  
=====
```

The Truman Show. 1998. 1h 47m. 19:30. Room 1
Genocidal Organ. 2017. 1hr 55m. 21:45. Room 1
The Mist. 2007. 2h 6m. 21:00. Room 2
Demon Slayer: Mugen Train. 2020. 1h59min. 23:20. Room 2
Mrs. Brown. 1997. 1h 41min. 19:00. Room 3
Peppa Pig: Festival of Fun. 2019. 1h 8min. 21:00. Room 3
Titanic. 1997. 3h 30min. 22:15. Room 3

Bye.

The time format must be correct. If the time entered is not valid, the program will print an error message and terminate the program

```
$ python3 pizzaz.py --show  
=====  
~~~~~  
~ Welcome to Pizzaz cinema ~  
~~~~~  
=====
```

Sorry. This program does not recognise the time format entered.

Bye.

```
$ python3 pizzaz.py --show abc  
=====  
~~~~~  
~ Welcome to Pizzaz cinema ~  
~~~~~  
=====
```

Sorry. This program does not recognise the time format entered.

Bye.

Hint: consider writing code to check if a time is valid or not. `str.isdigit()` can be helpful

Hint: consider converting all time to minutes for checking before after

After the output is shown, the program will go to the **End of the program**

Switch --book

To complete this, you would need to have completed at least week 3 lecture and tutorial

With this switch your program will process the order for 1 person.

Ask for the name of the movie

The program will ask the user to enter the name of the movie to watch:

```
What is the name of the movie you want to watch?
```

If the name of the movie does not exist, ask the user to try again, or quit the program.

```
Sorry, we could not find that movie. Enter Y to try again or N to quit.
```

For any answer other than `Y`, or `N`, the same question will be repeated.

When the answer is `N`, the program will go to the **End of the program**.

When answer is `Y` another prompt appears:

```
What is the name of the movie you want to watch?
```

It does not matter what time of day it is, the movie ticket order could be for the next day.

Once the movie is found, proceed to the next step.

Note: Movie name matching is case-insensitive, but must contain all the characters

Ask about popcorn!

Next, the program will ask the user if they would like a popcorn and if so, what size:

```
Would you like to order popcorn? Y/N
```

For any answer other than `Y`, or `N`, the same question will be repeated.

When answer `Y` another prompt appears:

```
You want popcorn. What size Small, Medium or Large? (S/M/L)
```

For any answer other than `S`, `M`, or `L`, the same question will be repeated.

Once a valid popcorn option is entered, proceed to the next step.

Generate and print the seat allocation

Next, you will generate the seat number for this person. For a single person, it is always seat number 17.

```
The seat number for person 1 is #17
```

For this switch, the next step is **Calculating price and itemise details**.

Switch --group

To complete this, you would need to have completed at least week 4 lecture and tutorial

With this switch your program will process the order for a group of people.

Ask for the name of the movie

The program will ask the user to enter the name of the movie to watch:

```
What is the name of the movie you want to watch?
```

If the name of the movie does not exist, ask the user to try again, or quit the program.

```
Sorry, we could not find that movie. Enter Y to try again or N to quit.
```

For any answer other than Y, or N, the same question will be repeated.

When the answer is N, the program will go to the **End of the program**.

When answer Y another prompt appears:

```
What is the name of the movie you want to watch?
```

Once the movie is found, proceed to the next step.

Note: It does not matter what time of day it is, the movie ticket order could be for the next day.

Note: Movie name matching is case-insensitive, but must contain all the characters

Ask for the number of people

Next, the program will ask the user to enter the number of people in the group:

```
How many persons will you like to book for?
```

You can always expect an integer number here. If the number of person is 1 or less, ask the user to try again, or quit the program.

Sorry, you must have at least two customers for a group booking. Enter Y to try again or N to quit.

For any answer other than Y, or N, the same question will be repeated.

When the answer is N, the program will go to the **End of the program**.

When the answer is Y, and the number of persons exceeds the capacity of the theater, print the error message and ask a further question:

Sorry, we do not have enough space to hold <X> people in the theater room of <Y> seats. Enter Y to try a different movie name or N to quit.

, where <X> is the number of persons entered and <Y> is the capacity of the room.

Capacity calculated as half the room size if it is an even number, or half the room size rounded up to the nearest integer if it is an odd number. Example:

- room size: 6 (even) -> capacity: 3. Maximum seating 1, 3, 5
- room size: 5 (odd) -> capacity: 3. Maximum seating 1, 3, 5

When the answer is Y, repeat from the earliest step in the --group switch asking for the movie name etc.

When the answer is N, the program will go to the **End of the program**.

For answers other than Y or N, the question is repeated.

Ask about popcorn for each person!

Next, the program will ask the user to enter if each person would like a popcorn and if so, what size:

For person 1, would you like to order popcorn? Y/N

When answer Y another prompt appears: Person 1 wants popcorn. What size
Small, Medium or Large? (S/M/L)

For every person, N persons, the same questions are repeated: For person 2, would you like to order popcorn? Y/N

...

For person <N>, would you like to order popcorn? Y/N

The answer options are to ordering popcorn for one person.

Generate and print the seat allocation

Next, you will generate the seat numbers based on the number of persons and the capacity of the room. Due to quarantine, every second seat is not available to sit in, as seat numbers must be spaced one apart.

The seat number for person 1 is #1

The seat number for person 2 is #3

The seat number for person 3 is #5

...

The seat number for person N is # $(2N-1)$

For this switch, the next step is **Calculating price and itemise details**.

Calculating price and itemise details

To complete this, you would need to have completed at least week 3 lecture and tutorial

In either case for a single person or a group of people, you will need to calculate price and itemise the details of the order.

The prices of tickets and popcorn will be as follows:

- Tickets are \$13.00 for movies starting before 16:00 and \$15.00 for movies starting from 16:00.
- Popcorn will cost \$3.50 for small, \$5.00 for medium and \$7.00 for large.

You will print the breakdown of the entire cost. Use the padding to format your output (see next section for formatting).

Example for one person (**--book** switch):

For 1 person, the initial cost is \$18.00

Person 1: Ticket before 16:00 \$13.00

Person 1: Medium popcorn \$ 5.00

No discounts applied \$ 0.00

The final price is \$18.00

Another example for one person (**--book** switch, no popcorn):

For 1 person, the initial cost is \$15.00

Person 1: Ticket from 16:00 \$15.00

```
No discounts applied          $ 0.00
```

```
The final price is           $15.00
```

Example for a group (--group switch):

```
For 5 persons, the initial cost is $87.00
```

```
Person 1: Ticket before 16:00  $13.00
```

```
Person 1: Medium popcorn       $ 5.00
```

```
Person 2: Ticket before 16:00  $13.00
```

```
Person 2: Medium popcorn       $ 5.00
```

```
Person 3: Ticket before 16:00  $13.00
```

```
Person 3: Medium popcorn       $ 5.00
```

```
Person 4: Ticket before 16:00  $13.00
```

```
Person 4: Small popcorn        $ 3.50
```

```
Person 5: Ticket before 16:00  $13.00
```

```
Person 5: Small popcorn        $ 3.50
```

```
No discounts applied          $ 0.00
```

```
The final price is           $87.00
```

There is an entry for discount. See the next section **Group Discount!** for understanding when and how it applies.

Formatting the output

Notice that the output of these examples align. To do this, you should consider using the `ljust` and `rjust` string functions.

```
person_counter = 4
```

```
ticket_time_type = 'before 16:00'
```

```
ticket_price = '13.00'
```

```
print(' Person {}: Ticket {}'.format(person_counter,
ticket_time_type).ljust(34, ' ') + '$' + '{}'.format(ticket_price).rjust(5,
' '))
```

```
person_counter = 4
```

```
popcorn_size = 'Large'
```

```
popcorn_price = '7.00'
```

```
print(' Person {}: {}  
popcorn'.format(person_counter,popcorn_size).ljust(34, ' ') + '$' +  
'{}'.format(popcorn_price).rjust(5, ' '))
```

For the switch option **--book**, the next step is **Transact and give change**.

For the switch option **--group**, the next step is **Group Discount!**.

Group Discount!

For the **--group** option only.

Discounts apply if the group purchase price exceeds \$100. The discount is automatically 10% off for each person's movie ticket and 20% off for each person's popcorn.

After discounts are applied, all final prices are rounded to the nearest 5 cent value.

- \$5.01 -> \$5.00
- \$5.02 -> \$5.00
- \$5.03 -> \$5.05
- \$5.04 -> \$5.05
- \$5.06 -> \$5.05
- \$5.07 -> \$5.05
- \$5.08 -> \$5.10
- \$5.09 -> \$5.10

Example for a group with a purchase of \$103.50:

For 6 persons, the initial cost is \$103.50

Person 1: Ticket before 16:00	\$13.00
Person 1: Medium popcorn	\$ 5.00
Person 2: Ticket before 16:00	\$13.00
Person 2: Medium popcorn	\$ 5.00
Person 3: Ticket before 16:00	\$13.00
Person 3: Medium popcorn	\$ 5.00
Person 4: Ticket before 16:00	\$13.00
Person 4: Small popcorn	\$ 3.50
Person 5: Ticket before 16:00	\$13.00
Person 5: Small popcorn	\$ 3.50
Person 6: Ticket before 16:00	\$13.00

Person 6: Small popcorn \$ 3.50

Discount applied tickets x6 -\$ 7.80

Discount applied popcorn x6 -\$ 5.10

The final price is \$90.60

Example for a group with a purchase of \$107.90 where there are persons without popcorn:

For 6 persons, the initial cost is \$107.00

Person 1: Ticket from 16:00 \$15.00

Person 1: Medium popcorn \$ 5.00

Person 2: Ticket from 16:00 \$15.00

Person 2: Medium popcorn \$ 5.00

Person 3: Ticket from 16:00 \$15.00

Person 4: Ticket from 16:00 \$15.00

Person 4: Small popcorn \$ 3.50

Person 5: Ticket from 16:00 \$15.00

Person 6: Ticket from 16:00 \$15.00

Person 6: Small popcorn \$ 3.50

Discount applied tickets x6 -\$ 9.00

Discount applied popcorn x4 -\$ 4.80

The final price is \$93.20

Warning: For floating point number calculations you should perform rounding to two decimal places *only* when printing the number. Please use `round(<x>, 2)` for that number.

Note: To accommodate the discount formatting, consider the following format string:

```
person_count = 6
```

```
popcorn_counter = 4
```

```
ticket_discount_print = round(9, 2)
```

```
popcorn_discount_print = round(4.8, 2)
```

```
print(' Discount applied tickets x{}'.format(person_count).ljust(33, ' ') +
      '-$' + '{:.2f}'.format(ticket_discount_print).rjust(5, ' '))
```

```
print(' Discount applied popcorn x{}'.format(popcorn_counter).ljust(33, ' ') +
      '-$' + '{:.2f}'.format(popcorn_discount_print).rjust(5, ' '))
```

For the switch option **--group**, the next step is **Transact and give change**.

Transact and give change

When a payment is required, you will need to prompt the user to pay. Ask for the amount of cash they will pay and calculate the change required. The change generated should be outputted to the user in forms of Australian currency denominations. These are:

- \$100
- \$50
- \$20
- \$10
- \$5
- \$2
- \$1
- 50c
- 20c
- 10c
- 5c

Note that your program should output the least number of notes/coins to make up the change. You may assume that the input will always be a positive floating point number with at most two decimal points.

If the user's payment is not enough, prompt them appropriately. See examples below.

Examples of Program Execution

Standard example:

For 1 person, the initial cost is \$18.50

Person 1: Ticket from 16:00 \$15.00

Person 1: Small popcorn \$ 3.50

No discounts applied \$ 0.00

The final price is \$18.50

Enter the amount paid: \$50.50

Change: \$32.00

\$20: 1

\$10: 1

\$ 2: 1

Not enough money:

For 1 person, the initial cost is \$18.50

Person 1: Ticket from 16:00	\$15.00
-----------------------------	---------

Person 1: Small popcorn	\$ 3.50
-------------------------	---------

No discounts applied	\$ 0.00
----------------------	---------

The final price is	\$18.50
--------------------	---------

Enter the amount paid: \$15

The user is \$3.50 short. Ask the user to pay the correct amount.

Enter the amount paid: \$20

Change: \$1.50

\$ 1: 1

50c: 1

User enters in an amount that isn't divisible by 5c:

For 1 person, the initial cost is \$18.50

Person 1: Ticket from 16:00	\$15.00
-----------------------------	---------

Person 1: Small popcorn	\$ 3.50
-------------------------	---------

No discounts applied	\$ 0.00
----------------------	---------

The final price is	\$18.50
--------------------	---------

Enter the amount paid: \$19.02

The input given is not divisible by 5c. Enter a valid payment.

Enter the amount paid: \$19.20

Change: \$0.70

50c: 1

20c: 1

The next step is **End of the program**

End of the program

Always print `Bye.` at the end of the program. Whether the order, or transaction was successful or not.

Tips and error cases to consider

- You should always print an empty line directly before asking the user for a command.
- switches, searches and answers to questions are case-insensitive. For example,
 - when answering a Y/N question, `Y` is valid, and `y` is valid.
 - when searching for a movie, `Fate/Stay Night: Heaven's Feel - III` is valid, and `Fate/staY Night: HEAVEN's Feel - iii` is also valid.
- when answering a question with fixed options, S/M/L or Y/N, if the answer is not one of these, repeat the question with the prompt indefinitely until the user enters a valid option.
- To print a string containing single quotes, enclose it in double quotes, and vice versa (or put a backslash before it).
- For the purposes of this assignment, you can treat all monetary values as floats - none of the test cases will cause floating point errors, however be mindful of rounding values when printing their result.

Getting started

- switch **--show** is the easiest. You need to perform a search of sorts and you should first place the movie data into variables, or even a list.
- switch **--book** is the next easiest, it requires careful flow control and calculation
- switch **--group** is the hardest and requires more consideration of flow control, calculation, lists and formatting.

Aim for the easy examples first. Things that you know are supposed to work.

Full examples

Show

```
$ python3 pizzaz.py --show 17:00
```

```
=====
```

```
~~~~~
```

```
~ Welcome to Pizzaz cinema ~
```



~~~~~  
=====

The Night Is Short, Walk on Girl. 2017. 1h 32m. 17:30. Room 1  
The Truman Show. 1998. 1h 47m. 19:30. Room 1  
Genocidal Organ. 2017. 1hr 55m. 21:45. Room 1  
Blade Runner 2049. 2017. 2h 44m. 17:45. Room 2  
The Mist. 2007. 2h 6m. 21:00. Room 2  
Demon Slayer: Mugen Train. 2020. 1h59min. 23:20. Room 2  
Soul. 2020. 1hr 40m. 17:00. Room 3  
Mrs. Brown. 1997. 1h 41min. 19:00. Room 3  
Peppa Pig: Festival of Fun. 2019. 1h 8min. 21:00. Room 3  
Titanic. 1997. 3h 30min. 22:15. Room 3

Bye.

## Book

\$ python3 pizzaz.py --book

=====

~~~~~

~ Welcome to Pizzaz cinema ~

~~~~~

=====

What is the name of the movie you want to watch? The Mist

Would you like to order popcorn? Y/N Y

You want popcorn. What size Small, Medium or Large? (S/M/L) G

You want popcorn. What size Small, Medium or Large? (S/M/L) L

The seat number for person 1 is #17

For 1 person, the initial cost is \$22.00

Person 1: Ticket from 16:00          \$15.00

|                         |         |
|-------------------------|---------|
| Person 1: Large popcorn | \$ 7.00 |
|-------------------------|---------|

|                      |         |
|----------------------|---------|
| No discounts applied | \$ 0.00 |
|----------------------|---------|

The final price is \$22.00

Enter the amount paid: \$36

Change: \$14.00

\$10: 1

\$ 2: 2

Bye.

Group 1

```
$ python3 pizzaz.py --group
```

```

=====
~
~
~ Welcome to Pizzaz cinema ~
~
~
=====

```

What is the name of the movie you want to watch? The Matrix

How many persons will you like to book for? 3

For person 1, would you like to order popcorn? Y/N Y

Person 1 wants popcorn. What size Small, Medium or Large? (S/M/L) L

For person 2, would you like to order popcorn? Y/N Y

Person 2 wants popcorn. What size Small, Medium or Large? (S/M/L) S

For person 3, would you like to order popcorn? Y/N Y

Person 3 wants popcorn. What size Small, Medium or Large? (S/M/L) S

The seat number for person 1 is #1

The seat number for person 2 is #3

The seat number for person 3 is #5

For 3 persons, the initial cost is \$53.00

Person 1: Ticket before 16:00      \$13.00

|                         |         |
|-------------------------|---------|
| Person 1: Large popcorn | \$ 7.00 |
|-------------------------|---------|

```
Person 2: Ticket before 16:00    $13.00
```

|                         |         |
|-------------------------|---------|
| Person 2: Small popcorn | \$ 3.50 |
|-------------------------|---------|

Person 3: Ticket before 16:00      \$13.00

|                         |         |
|-------------------------|---------|
| Person 3: Small popcorn | \$ 3.50 |
|-------------------------|---------|

|                      |         |
|----------------------|---------|
| No discounts applied | \$ 0.00 |
|----------------------|---------|

The final price is \$53.00

Enter the amount paid: \$100

Change: \$47.00

\$20 : 2

\$ 5: 1

\$ 2: 1

Bye.

Expand (40 lines)

Group 2

```
$ python3 pizzaz.py --group
```

```

=====
~ ~ ~ ~ ~
~ Welcome to Pizzaz cinema ~
~ ~ ~ ~ ~
=====

```

What is the name of the movie you want to watch? The Matrix

How many persons will you like to book for? 6

For person 1, would you like to order popcorn? Y/N Y

Person 1 wants popcorn. What size Small, Medium or Large? (S/M/L) M

For person 2, would you like to order popcorn? Y/N Y

Person 2 wants popcorn. What size Small, Medium or Large? (S/M/L) M

For person 3, would you like to order popcorn? Y/N Y

Person 3 wants popcorn. What size Small, Medium or Large? (S/M/L) M

For person 4, would you like to order popcorn? Y/N Y

Person 4 wants popcorn. What size Small, Medium or Large? (S/M/L) S

For person 5, would you like to order popcorn? Y/N Y

Person 5 wants popcorn. What size Small, Medium or Large? (S/M/L) S

For person 6, would you like to order popcorn? Y/N Y

Person 6 wants popcorn. What size Small, Medium or Large? (S/M/L) S

The seat number for person 1 is #1

The seat number for person 2 is #3

The seat number for person 3 is #5

The seat number for person 4 is #7

The seat number for person 5 is #9

The seat number for person 6 is #11

For 6 persons, the initial cost is \$103.50

|                               |         |
|-------------------------------|---------|
| Person 1: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                          |         |
|--------------------------|---------|
| Person 1: Medium popcorn | \$ 5.00 |
|--------------------------|---------|

|                               |         |
|-------------------------------|---------|
| Person 2: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                          |         |
|--------------------------|---------|
| Person 2: Medium popcorn | \$ 5.00 |
|--------------------------|---------|

|                               |         |
|-------------------------------|---------|
| Person 3: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                          |         |
|--------------------------|---------|
| Person 3: Medium popcorn | \$ 5.00 |
|--------------------------|---------|

|                               |         |
|-------------------------------|---------|
| Person 4: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                         |         |
|-------------------------|---------|
| Person 4: Small popcorn | \$ 3.50 |
|-------------------------|---------|

|                               |         |
|-------------------------------|---------|
| Person 5: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                         |         |
|-------------------------|---------|
| Person 5: Small popcorn | \$ 3.50 |
|-------------------------|---------|

|                               |         |
|-------------------------------|---------|
| Person 6: Ticket before 16:00 | \$13.00 |
|-------------------------------|---------|

|                         |         |
|-------------------------|---------|
| Person 6: Small popcorn | \$ 3.50 |
|-------------------------|---------|

```
Discount applied tickets x6      -$ 7.80
Discount applied popcorn x6      -$ 5.10
```

```
The final price is                $90.60
```

```
Enter the amount paid: $120
```

```
Change: $29.40
```

```
$20: 1
```

```
$ 5: 1
```

```
$ 2: 2
```

```
20c: 2
```

```
Bye.
```

Expand (57 lines)

### Group 3

```
$ python3 pizzaz.py --group
```

```
=====
~
~
~ Welcome to Pizzaz cinema ~
~
~
~
=====
```

```
What is the name of the movie you want to watch? The Shining
```

```
How many persons will you like to book for? 36
```

```
Sorry, we do not have enough space to hold 36 people in the theater room of
17 seats. Enter Y to try a different movie name or N to quit.Y
```

```
What is the name of the movie you want to watch? The Shining
```

```
How many persons will you like to book for? 2
```

```
For person 1, would you like to order popcorn? Y/N N
```

For person 2, would you like to order popcorn? Y/N N

The seat number for person 1 is #1

The seat number for person 2 is #3

For 2 persons, the initial cost is \$26.00

Person 1: Ticket before 16:00      \$13.00

Person 2: Ticket before 16:00      \$13.00

No discounts applied                      \$ 0.00

The final price is                              \$26.00

Enter the amount paid: \$30.30

Change: \$4.30

\$ 2: 2

20c: 1

10c: 1

Bye.

Expand (36 lines)

## Code Submission

Your code submission must be made via Ed in this lesson.

- To make a submission, you will need to press the "Mark" button.
- You may submit as many times as you wish without penalty.
- You are able to view previous submissions from the code submission section.
- Every submission you make includes the `README.md` and the `pizzaz.py` file

The following rules apply to your code submission:

- Only the file `pizzaz.py` will be graded by the automarker
- Only the last submission will be graded.
  - Only the final `README.md` file of the last submission will be graded by your tutor
- Submission after the due date will incur a late penalty as described in the unit of study outline.

- Your submission must be able to compile and run within the Ed environment provided.

After each submission, the marking system will automatically check your code against the public test cases.

The Python version is that which is presently being used on Ed system:

```
$ python3 --version  
Python 3.9.2
```

Please ensure you carefully follow the assignment specification. Your program output must exactly match the output shown in the examples.

## Late is late!

The computer does not discriminate about what is late. There are two files for submission

- pizzaz.py
- README.md

They are both in your Ed workspace. Make sure these are the ones for your submission.

Changes to these files after the deadline will incur a penalty for both (not individually).

## Marking criteria

This assignment is 15% of your final course grade.

The grade you will receive is based on the number of test cases passed as well as manual grading by your tutor.

We have provided you with some sample test cases but these do not test all the functionality described in the assignment. It is important that you thoroughly test your own code.

### Automatic tests 10 / 15

These are marked by a computer.

- There are public test cases
- There are hidden test cases
- There are private test cases

### Manual grading 5 / 15

The manual grading from your tutor will consider the style, layout and comments you provide in your code. Your tutor will not be debugging or running your code against any test cases.

- edit the README.md file to describe the parts you have completed and the general structure of your code. Maximum 500 words. See scaffold file directory in Ed.
  - describe how you store and access the movie data
  - describe how you have done error checking on the time data and how you have compared two different times
  - list each of the idioms "search tasks" you have used in your solution
- in your code file pizzaz.py:
  - Please ensure your code is comprehensible for the purposes of marking. We recommend that your code adheres to the PEP 8 style guide <https://www.python.org/dev/peps/pep-0008/>
  - Please use comments or docstrings for sections of code to explain the section's purpose. e.g. the following block of code will "search for", "calculate", "get input for options" etc.

## Academic declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy, and except where specifically acknowledged, the work contained in this assignment or project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the the Academic Dishonesty and Plagiarism in Coursework Policy, can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Computer Science, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and or communicate a copy of this assignment to a plagiarism checking service or in house computer program, and that a copy of the assignment may be maintained by the service or the School of Computer Science for the purpose of future plagiarism checking.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not



properly follow the assignment description, or your code is unnecessarily or deliberately obfuscated.