Write a class called RestaurantCheck. It should have the following: (use OOPs concepts)

- Fields called check_number, sales_tax_percent, subtotal, table_number, and server_name representing an identification for the check, the bill without tax added, the sales tax percentage, the table number, and the name of the server.
- A constructor that sets the values of all four fields
- A method called calculate_total that takes no arguments (besides self) and returns the total bill including sales tax.
- A method called print_check that writes to a file called check###.txt, where ### is the check number and writes information about the check to that file, formatted like below:

```
Check Number: 443
Sales tax: 6.0%
Subtotal: $23.14
Total: $24.53
Table Number: 17
Server: Sonic the Hedgehog
```

Test the class by creating a RestaurantCheck object and calling the print_check() method.

In [2]:
```
class RestaurantCheck:
    def __init__(self, check_number, sales_tax_percent, subtotal, table_number, server_name):
        self.check_number = check_number
        self.sales_tax_percent = sales_tax_percent
        self.subtotal = subtotal
        self.table_number = table_number
        self.server_name = server_name

    def calculate_total(self):
        total = self.subtotal + (self.subtotal * self.sales_tax_percent / 100)
        return total

    def print_check(self):
        filename = f"check{self.check_number}.txt"
        with open(filename, "w") as file:
            file.write(f"Check Number: {self.check_number}\n")
            file.write(f"Sales tax: {self.sales_tax_percent}%\n")
            file.write(f"Subtotal: ${self.subtotal:.2f}\n")
            file.write(f"Total: ${self.calculate_total():.2f}\n")
            file.write(f"Table Number: {self.table_number}\n")
            file.write(f"Server: {self.server_name}\n")
        print(f"Check saved as {filename}.")

check = RestaurantCheck(443, 6.0, 23.14, 17, "Sonic the Hedgehog")
check.print_check()
```

```
Check saved as check443.txt.
```

Write a Regular Expression Python function to Validate

Phone No, (Must be 10 digits)
Name, (first Char must be uppercase)
E-Mail, (abc@abc.com)
Date .(DD-MM-YYYY)

In [10]:
```
phone_number = "8328335185"
name = "Lakshmi"
email = "venkatalakshmimorla123@gmail.com"
date = "12-06-2023"

if validate_input(8328335185, Lakshmi, venkatalakshmimorla123@gmail.com, 12-6-2023):
    print("All inputs are valid.")
else:
    print("Invalid inputs detected.")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[10], line 6
      3 email = "venkatalakshmimorla123@gmail.com"
      4 date = "12-06-2023"
----> 6 if validate_input(8328335185, Lakshmi, venkatalakshmimorla123@gmail.com, 12-6-2023):
      7     print("All inputs are valid.")
      8 else:

NameError: name 'validate_input' is not defined
```

In [ ]:

In [ ]:

In [ ]: