

Unit: CFC011023

Python Fundamentals

Student No: S12

Student Name: Ivan Wong Sen Loong

Trainer's Name: Mr. Tushar

Date of completion: 30-01-2024

Table of Contents

1. Contact Management System	2
2. To do List Application.....	4
3. Palindrome Checker	6
4. Calculator	7
5. File searcher	8
6. Task Scheduler	9
7. Online Shopping System	10

1. Contact Management System

```
contacts = []

def add_contact():
    name = input("Enter the name of the contact: ")
    phone = input("Enter the phone number: ")
    email = input("Enter the email address: ")
    contacts.append({'name': name, 'phone': phone, 'email': email})

def list_contacts():
    for index, contact in enumerate(contacts, 1):
        print(f"{index}. Name: {contact['name']}, Phone: {contact['phone']}, Email: {contact['email']}")

def search_contact():
    search_name = input("Enter the name to search for: ")
    for contact in contacts:
        if search_name.lower() in contact['name'].lower():
            print(f"Name: {contact['name']}, Phone: {contact['phone']}, Email: {contact['email']}")
            return
    print("Contact not found.")

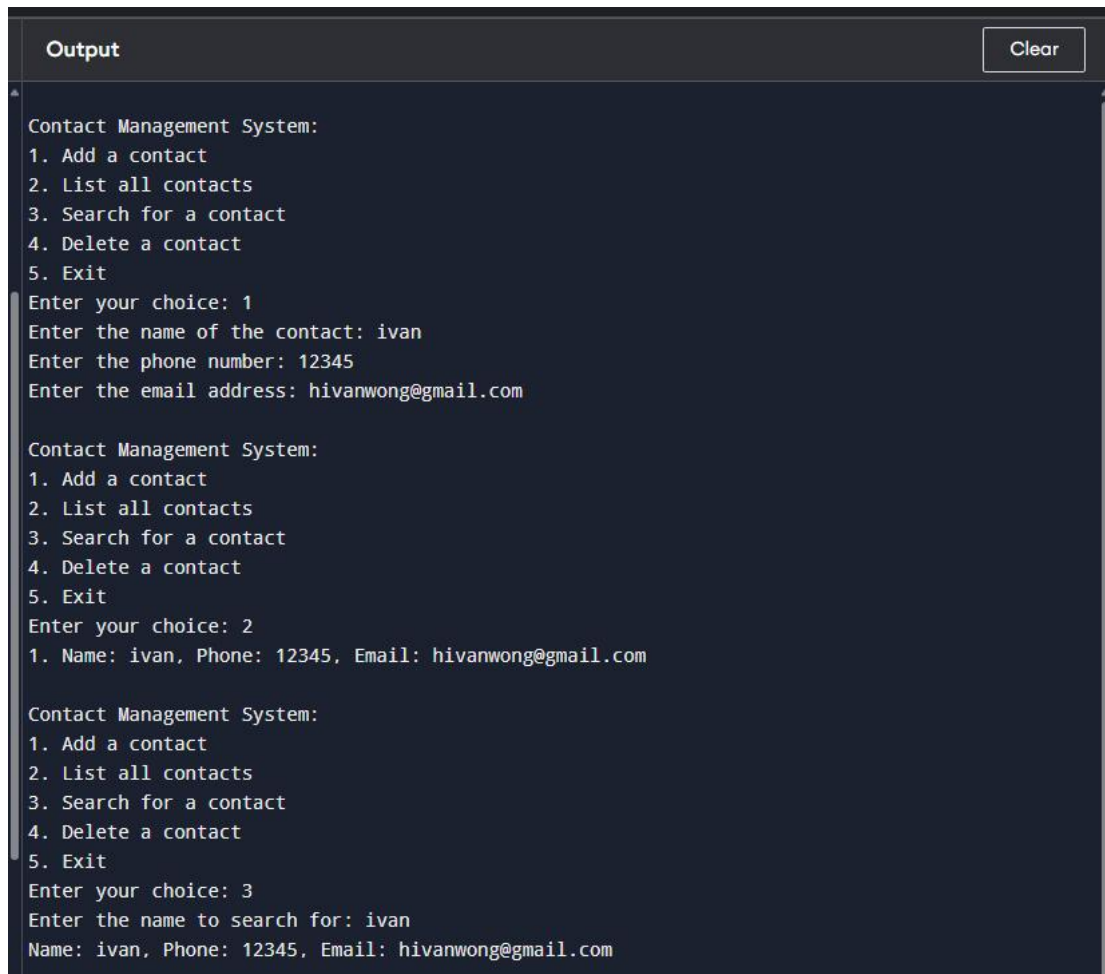
def display_menu():
    print("\nContact Management System:")
    print("1. Add a contact")
    print("2. List all contacts")
    print("3. Search for a contact")
    print("4. Delete a contact")
    print("5. Exit")
    choice = input("Enter your choice: ")
    return int(choice)

def delete_contact():
    del_name = input("Enter the name of the contact to delete: ")
    for contact in contacts:
        if del_name.lower() == contact['name'].lower():
            contacts.remove(contact)
            print(f"Deleted contact {del_name}.")
            return
    print("Contact not found.")

while True:
    choice = display_menu()

    if choice == 1:
        add_contact()
    elif choice == 2:
        list_contacts()
    elif choice == 3:
        search_contact()
    elif choice == 4:
        ## MISSING_FUNCTION
    elif choice == 5:
        print("Exiting the system. Goodbye!")
```

```
        break
    else:
        print("Invalid choice. Please choose again.")
```



```
Output
Contact Management System:
1. Add a contact
2. List all contacts
3. Search for a contact
4. Delete a contact
5. Exit
Enter your choice: 1
Enter the name of the contact: ivan
Enter the phone number: 12345
Enter the email address: hivanwong@gmail.com

Contact Management System:
1. Add a contact
2. List all contacts
3. Search for a contact
4. Delete a contact
5. Exit
Enter your choice: 2
1. Name: ivan, Phone: 12345, Email: hivanwong@gmail.com

Contact Management System:
1. Add a contact
2. List all contacts
3. Search for a contact
4. Delete a contact
5. Exit
Enter your choice: 3
Enter the name to search for: ivan
Name: ivan, Phone: 12345, Email: hivanwong@gmail.com
```

2. To-do List Application

```
todos = []

def add_todo():
    task = input("Enter your task: ")
    todos.append(task)

def display_todos():
    for idx, todo in enumerate(todos, 1):
        print(f"{idx}. {todo}")

def delete_todo():
    display_todos()
    idx = int(input("Enter task number to delete: "))
    if 0 < idx <= len(todos):
        del todos[idx-1]
    else:
        print("Invalid index!")

def main():
    while True:
        print("\n1. Add Task\n2. Display Tasks\n3. Delete Task\n4. Exit")
        choice = input("Enter your choice: ")
        if choice == "1":
            add_todo()
        elif choice == "2":
            display_todos()
        elif choice == "3":
            delete_todo()
        elif choice == "4":
            break
        else:
            print("Invalid choice!")

main()
```

Output

Clear

```
1. Add Task
2. Display Tasks
3. Delete Task
4. Exit
Enter your choice: 1
Enter your task: do work

1. Add Task
2. Display Tasks
3. Delete Task
4. Exit
Enter your choice: 1
Enter your task: eat pizza

1. Add Task
2. Display Tasks
3. Delete Task
4. Exit
Enter your choice: 1
Enter your task: sleep

1. Add Task
2. Display Tasks
3. Delete Task
4. Exit
Enter your choice: 2
1. do work
2. eat pizza
3. sleep
```

3. Palindrome Checker

```
def is_palindrome(string):  
    return string == string[::-1]  
  
def main():  
    while True:  
        word = input("Enter a word to check if it's a palindrome: ")  
        if is_palindrome(word):  
            print(f"{word} is a palindrome!")  
            break # Exit loop once palindrome is found  
        else:  
            print(f"{word} is not a palindrome! Try again.\n")  
  
main()
```

Output Clear

```
Enter a word to check if it's a palindrome: test  
test is not a palindrome! Try again.  
  
Enter a word to check if it's a palindrome: madam  
madam is a palindrome!  
  
=== Code Execution Successful ===
```

4. Calculator

```
def add(x, y):
    return x + y

def subtract(x, y):
    Return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y == 0:
        return "Undefined (division by zero)"
    return x / y

def main():
    while True:
        print("\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. Exit")
        choice = input("Enter your choice: ")

        if choice == "5":
            break

        x = float(input("Enter first number: "))
        y = float(input("Enter second number: "))

        if choice == "1":
            print(f"Result: {add(x, y)}")
        elif choice == "2":
            print(f"Result: {subtract(x, y)}")
        elif choice == "3":
            print(f"Result: {multiply(x, y)}")
        elif choice == "4":
            print(f"Result: {divide(x, y)}")
        else:
            print("Invalid choice!")

main()
```

1. Add 2. Subtract 3. Multiply 4. Divide 5. Exit Enter your choice: 1 Enter first number: 1 Enter second number: 2 Result: 3.0	1. Add 2. Subtract 3. Multiply 4. Divide 5. Exit Enter your choice: 2 Enter first number: 4 Enter second number: 2 Result: 2.0	1. Add 2. Subtract 3. Multiply 4. Divide 5. Exit Enter your choice: 3 Enter first number: 4 Enter second number: 5 Result: 20.0	1. Add 2. Subtract 3. Multiply 4. Divide 5. Exit Enter your choice: 4 Enter first number: 5 Enter second number: 2 Result: 2.5
--	--	---	--

5. File searcher

```
# File Searcher

import os

def find_large_files(directory, size_limit):
    large_files = []

    for foldername, subfolders, filenames in os.walk(directory):
        for filename in filenames:
            filepath = os.path.join(foldername, filename)
            file_size = os.path.getsize(filepath)

            if file_size > size_limit:
                large_files.append(filepath)

    return large_files

def main():
    directory_to_search = input("Enter the directory path to search: ")

    size_limit = 100 * 1024 * 1024

    result = find_large_files(directory_to_search, size_limit)

    if result:
        print(f"Files larger than ### MB in '{directory_to_search}':")
        for file_path in result:
            print(file_path)
    else:
        print(f"No files larger than ### MB found in '{directory_to_search}'.")
```


6. Task Scheduler

```
import time

tasks = []

def add_task():
    task = input("Enter your task: ")
    duration = int(input("Enter duration in seconds to wait: "))
    tasks.append((task, duration))

def start_tasks():
    for task, duration in tasks:
        print(f"Starting task: {task}")
        time.sleep(duration)
        print(f"Completed task: {task}")

def main():
    while True:
        print("\n1. Add Task\n2. Start Tasks\n3. Exit")
        choice = input("Enter your choice: ")
        if choice == "1":
            add_task()
        elif choice == "2":
            start_tasks()
        elif choice == "3":
            break
        else:
            print("Invalid choice!")

main()
```

```
1. Add Task
2. Start Tasks
3. Exit
Enter your choice: 1
Enter your task: Do work
Enter duration in seconds to wait: 30

1. Add Task
2. Start Tasks
3. Exit
Enter your choice: 2
Starting task: Do work
Completed task: Do work

1. Add Task
2. Start Tasks
3. Exit
Enter your choice: |
```

7. Online Shopping System

```
cart = []

def display_products(products):
    print("\nAvailable Products:")
    for idx, product in enumerate(products, 1):
        print(f"{idx}. {product['name']} - ${product['price']}")

def add_to_cart(product_idx, products):
    if 0 < product_idx <= len(products):
        cart.append(products[product_idx-1])
        print(f"{products[product_idx-1]['name']} added to cart!")
    else:
        print("Invalid product index!")

def display_cart():
    print("\nYour Cart:")
    total_price = 0
    for item in cart:
        print(f"{item['name']} - ${item['price']}")
        total_price += item['price']
    print(f"Total: ${total_price:.2f}")

def apply_discount(total, discount_rate):
    return total * (1 - discount_rate/100)

def main():
    products = [
        {"name": "T-shirt", "price": 19.99},
        {"name": "Jeans", "price": 49.99},
        {"name": "Sneakers", "price": 89.99},
        {"name": "Hat", "price": 14.99}
    ]

    while True:
        display_products(products)
        choice = input("\nEnter product number to add to cart or 'c' to checkout or 'q' to quit: ")

        if choice.isdigit():
            add_to_cart(int(choice), products)
        elif choice == 'c':
            display_cart()
            total = sum(item['price'] for item in cart)
            discounted_total = apply_discount(total, 10)
            print(f"Total after discount: ${discounted_total:.2f}")
        elif choice == 'q':
            print("Thanks for shopping with us!")
            break
        else:
            print("Invalid choice!")

main()
```

```
Available Products:
1. T-shirt - $19.99
2. Jeans - $49.99
3. Sneakers - $89.99
4. Hat - $14.99

Enter product number to add to cart or 'c' to checkout or 'q' to quit: 2
Jeans added to cart!

Available Products:
1. T-shirt - $19.99
2. Jeans - $49.99
3. Sneakers - $89.99
4. Hat - $14.99

Enter product number to add to cart or 'c' to checkout or 'q' to quit: 3
Sneakers added to cart!

Available Products:
1. T-shirt - $19.99
2. Jeans - $49.99
3. Sneakers - $89.99
4. Hat - $14.99

Enter product number to add to cart or 'c' to checkout or 'q' to quit: c

Your Cart:
Jeans - $49.99
Sneakers - $89.99
Total: $139.98
Total after discount: $125.98
```