```
"""
population_increases.py
@author: ITP 150
Date Created: Oct. 5, 2024
https://unicode.org/emoji/charts/full-emoji-list.html
"""


import csv
import os


def main():
    DISPLAY_LIST = 1
    STATS = 2
    SAVE_STATS = 3
    QUIT = 99
    pop_list = []
    pop_stats = {}
    choice = 0
    if not os.path.isfile('us_population.csv'):
        print('Sorry we can\'t run the program at this time.')
    else:
        pop_list = read_the_file(pop_list)
    if pop_list:
        # print(pop_list)  # a sanity check to see if we read the data
        while choice != QUIT:
            menu_string = 'Please choose from the following menu: \
            \nEnter 1 to print the pop list.\
            \nEnter 2 to analyze pop statistics.\
            \nEnter 3 to save the statistics.\
            \nEnter 99 to Quit.'
            valid_choices = [1, 2, 3, 99]
            choice = input_menu_choice(menu_string, valid_choices)
            if choice == DISPLAY_LIST:
                display_list(pop_list)
            elif choice == STATS:
                pop_stats = calc_descriptive_stats(pop_list, pop_stats)
            elif choice == SAVE_STATS:
                save_stats(pop_stats)
            else:
                print('Thanks for using the US population tracker.')


def read_the_file(pop_list):
    try:
        with open('us_population.csv', newline='') as pop_file:
            pop_reader = csv.reader(pop_file, delimiter=',')  # \t tab
            pop_list = [row for row in pop_reader]
        for row in range(1, len(pop_list)):
            pop_list[row][0] = int(pop_list[row][0])
            pop_list[row][1] = int(pop_list[row][1])

        return pop_list
    except IOError:
        print('An error occurred trying to read from the file.')
    except IndexError:
        print('An index error occurred.')
    except Exception:
        print('An error occurred.')


def input_menu_choice(menu_string, valid_choices):
```

```
    while True:
        try:
            print('-'*50)
            print(menu_string)
            print('-'*50)
            # Get the user's choice
            choice = int(input())
            if choice in valid_choices:
                return choice
            else:
                raise Exception
        except ValueError:  # throws on invalid data type
            print('Invalid choice. Please try again: \U0001F600')
        except Exception:  # throws on anything else
            print('Valid datatype but invalid value. Please try again.')


def display_list(pop_list):
    print('='*35)
    print(f'{"As Of Date":12s}{"pop":>20s}')
    try:
        for row in range(1, len(pop_list)):
            print(f'{pop_list[row][0]:12}{pop_list[row][1]:>20,d}')
        print('='*35)
    except IndexError:
        print('Index is out of range.')
    except Exception:
        print('An error occurred.')


def calc_descriptive_stats(pop_list, pop_stats):
    average_pop = calc_average(pop_list)
    lowest_pop, lowest_year = find_lowest(pop_list)
    highest_pop, highest_year = find_highest(pop_list)
    try:
        labels = ['Average Population', 'Lowest Population', 'Lowest Year',
                  'Highest Population', 'Highest Year']
        stats = [average_pop, lowest_pop, lowest_year, highest_pop,
                 highest_year]
        stats_zipped = zip(labels, stats)
        pop_stats = dict(stats_zipped)
        for key, val in pop_stats.items():
            print(f'{key:30s}{val:>20,.0f}')
        return pop_stats
    except KeyError:
        print("A key error occurred.")
    except Exception:
        print('An error occurred.')


def calc_average(pop_list):
    sum_pop = 0
    try:
        for row in range(1, len(pop_list)):
            sum_pop = sum_pop + pop_list[row][1]
        average_pop = sum_pop / (len(pop_list) - 1)
        # print(average_pop)
        return average_pop
    except IndexError:
        print('Index is out of range')
    except ZeroDivisionError:
        print('A Zero Division Error occurred.')
    except Exception:
```

```
        print('An error occurred.')


def find_lowest(pop_list):
    try:
        # print(pop_list)
        lowest_pop = pop_list[1][1]
        lowest_year = pop_list[1][0]
        for row in range(1, len(pop_list)):
            if pop_list[row][1] < lowest_pop:
                lowest_pop = pop_list[row][1]
                lowest_year = pop_list[row][0]
        return lowest_pop, lowest_year
    except IndexError:
        print('An Index Error Occurred.')
    except Exception:
        print('An error occurred.')


def find_highest(pop_list):
    try:
        highest_pop = pop_list[1][1]
        highest_year = pop_list[1][0]
        for row in range(1, len(pop_list)):
            if pop_list[row][1] > highest_pop:
                highest_pop = pop_list[row][1]
                highest_year = pop_list[row][0]
        return highest_pop, highest_year
    except IndexError:
        print('An Index Error Occurred.')
    except Exception:
        print('An error occurred.')


def save_stats(pop_stats):
    try:
        pop_stats_file = csv.writer(open('pop_stats.csv', 'w'))
        for key, val in pop_stats.items():
            pop_stats_file.writerow([key, val])
        print("The pop_stats_file.csv has been updated.")
    except IOError:
        print("An error occurred trying to write to the file.")
    except KeyError:
        print("A key error occurred.")
    except Exception:
        print('An error occurred.')


if __name__ == '__main__':
    main()

```