```python
from os.path import exists
from json import dump, load


def main() -> None:
    splash_screen()
    while True:
        selection = check_option(1, 11, "Select an action: ")
        match selection:
            case 1:
                print("Enter the name of the class you wish to view.")
                class_name = input("=> ")
                view_class(class_name)
            case 2:
                list_class()
            case 3:
                add_class()
            case 4:
                print("Which class would you like to expunge?")
                class_name = input("=> ").lower()
                remove_class(class_name)
            case 5:
                print("Which class would you like to modify?")
                class_name = input("=> ").lower()
                modify_grade(class_name)
            case 6:
                print("Enter the year of classes you wish to view")
                year = check_option(1, 10_000)
                print(
                    """
Which GPA would you like to view?
1\tUnweighted
2\tWeighted
                    """
                )
                weighted = check_option(1, 3) == 2
                print(f"{year_gpa(year, weighted):.2f}")
            case 7:
                print(
                    """
Which GPA would you like to view?
1\tUnweighted
2\tWeighted
                    """
                )
                weighted = check_option(1, 3) == 2
                print(f"{cumulative_gpa(weighted):.2f}")
            case 8:
                check_college(cumulative_gpa())
            case 9:
                raise SystemExit("Thank you for using the program")
            case 10:
                splash_screen()


def splash_screen():
    print(
        """
1\tView a class
2\tView class history
3\tAdd a class
4\tExpunge a class
5\tModify class grade
6\tView your GPA for a particular year
7\tView your cumulative GPA
8\tView the colleges you can attend with your GPA (Virginia)
9\tQuit the program
10\tRepeat this message
        """
    )


def check_option(lower_bounds: int, upper_bounds: int, prompt: str = "=> ") -> int:
    while True:
        user_input = input(prompt)
        if user_input.isdigit():
            user_input = int(user_input)
            if user_input in range(lower_bounds, upper_bounds):
                return user_input
        print("Please enter a valid option.")

```

```python

def read_gpa() -> dict:
    path = "../resources/gpa.json"
    if not exists(path):
        write_gpa({"class": {}, "total classes": 0})
    with open(path) as gpa_file:
        return load(gpa_file)


def write_gpa(data: dict) -> None:
    with open("../resources/gpa.json", "w") as gpa_file:
        dump(data, gpa_file)


def letter_to_gpa(letter: str, class_type: str, weighted: bool = False) -> float:
    match letter:
        case "A":
            gpa = 4.0
        case "A-":
            gpa = 3.7
        case "B+":
            gpa = 3.3
        case "B":
            gpa = 3.0
        case "B-":
            gpa = 2.7
        case "C+":
            gpa = 2.3
        case "C":
            gpa = 2.0
        case "C-":
            gpa = 1.7
        case "D+":
            gpa = 1.3
        case "D":
            gpa = 1.0
        case "F":
            gpa = 0.0
    if gpa != 0.0 and weighted:
        match class_type:
            case "AP" | "DE":
                gpa += 1.0
            case "HN":
                gpa += 0.5
    return gpa


def add_class():
    gpa_data = read_gpa()
    print("What is the name of your class?")
    class_name = input("=> ").lower()
    print(
        """
What is your letter grade?
1\tA
2\tA-
3\tB+
4\tB
5\tB-
6\tC+
7\tC
8\tC-
9\tD+
10\tD
11\tF
        """
    )
    selection = check_option(1, 12)
    match selection:
        case 1:
            letter_grade = "A"
        case 2:
            letter_grade = "A-"
        case 3:
            letter_grade = "B+"
        case 4:
            letter_grade = "B"
        case 5:
            letter_grade = "B-"
        case 6:
            letter_grade = "C+"
```

```
163         case 7:
164             letter_grade = "C"
165         case 8:
166             letter_grade = "C-"
167         case 9:
168             letter_grade = "D+"
169         case 10:
170             letter_grade = "D"
171         case 11:
172             letter_grade = "F"
173     print(
174         """
175 What type of class are you in?
176 1\tAdvanced Placement
177 2\tDual Enrollment
178 3\tHonors
179 4\tGeneral Education
180         """
181     )
182     selection = check_option(1, 5)
183     match selection:
184         case 1:
185             class_type = "AP"
186         case 2:
187             class_type = "DE"
188         case 3:
189             class_type = "HN"
190         case 4:
191             class_type = "GE"
192     print("What year did you take this class?")
193     year = check_option(0, 10_000)
194 
195     gpa = letter_to_gpa(letter_grade, class_type)
196     weighted_gpa = letter_to_gpa(letter_grade, class_type, weighted=True)
197     gpa_data["class"][class_name] = {
198         "grade": letter_grade,
199         "type": class_type,
200         "year": year,
201         "gpa": gpa,
202         "weighted gpa": weighted_gpa
203     }
204     gpa_data["total classes"] += 1
205     write_gpa(gpa_data)
206 
207 
208 def remove_class(class_name: str) -> None:
209     gpa_data = read_gpa()
210     x = gpa_data["class"].pop(class_name)
211     if x:
212         gpa_data["total classes"] -= 1
213     write_gpa(gpa_data)
214 
215 
216 def modify_grade(class_name: str):
217     gpa_data = read_gpa()
218     for key in gpa_data["class"]:
219         if key == class_name:
220             print(
221                 """
222 What is your letter grade?
223 1\tA
224 2\tA-
225 3\tB+
226 4\tB
227 5\tB-
228 6\tC+
229 7\tC
230 8\tC-
231 9\tD+
232 10\tD
233 11\tF
234                 """
235             )
236             selection = check_option(1, 12)
237             match selection:
238                 case 1:
239                     gpa_data["class"][key]["grade"] = "A"
240                 case 2:
241                     gpa_data["class"][key]["grade"] = "A-"
242                 case 3:
243                     gpa_data["class"][key]["grade"] = "B+"
```

```
                case 4:
                    gpa_data["class"][key]["grade"] = "B"
                case 5:
                    gpa_data["class"][key]["grade"] = "B-"
                case 6:
                    gpa_data["class"][key]["grade"] = "C+"
                case 7:
                    gpa_data["class"][key]["grade"] = "C"
                case 8:
                    gpa_data["class"][key]["grade"] = "C-"
                case 9:
                    gpa_data["class"][key]["grade"] = "D+"
                case 10:
                    gpa_data["class"][key]["grade"] = "D"
                case 11:
                    gpa_data["class"][key]["grade"] = "F"
            gpa_data["class"][key]["gpa"] = letter_to_gpa(
                gpa_data["class"][key]["grade"], gpa_data["class"][key]["grade"])
            gpa_data["class"][key]["weighted gpa"] = letter_to_gpa(
                gpa_data["class"][key]["grade"], gpa_data["class"][key]["grade"], True)
            write_gpa(gpa_data)


def list_class() -> None:
    gpa_data = read_gpa()["class"]
    print()
    for key in sorted(gpa_data.keys()):
        print(key)
    print()


def view_class(class_name):
    gpa_data = read_gpa()["class"]
    for key in gpa_data:
        if key == class_name:
            print(
                f"""
{key}
    Grade: {gpa_data[key]['grade']}
    Type: {gpa_data[key]['type']}
    Year: {gpa_data[key]['year']}
    GPA (Unweighted): {gpa_data[key]['gpa']}
    GPA (Weighted): {gpa_data[key]['weighted gpa']}
                """
            )


def cumulative_gpa(weighted: bool = False) -> float:
    gpa_data = read_gpa()
    total_gpa = 0
    total_weighted_gpa = 0
    for key in gpa_data["class"]:
        total_gpa += gpa_data["class"][key]["gpa"]
        total_weighted_gpa += gpa_data["class"][key]["weighted gpa"]
    if weighted:
        return total_weighted_gpa / gpa_data["total classes"]
    else:
        return total_gpa / gpa_data["total classes"]


def year_gpa(year: int, weighted: bool = False) -> float:
    gpa_data = read_gpa()["class"]
    total_gpa = 0
    total_weighted_gpa = 0
    classes_in_year = 0
    for key in gpa_data:
        if gpa_data[key]["year"] == year:
            total_gpa += gpa_data[key]["gpa"]
            total_weighted_gpa += gpa_data[key]["weighted gpa"]
            classes_in_year += 1
    if classes_in_year == 0:
        return 0
    if weighted:
        return total_weighted_gpa / classes_in_year
    else:
        return total_gpa / classes_in_year


def check_college(gpa):
    # Data from https://www.ubridge.org/ubridgepvcc/guaranteed-admissions-universities/
    colleges = {
```

```
        "University of Virginia": 3.4,
        "College of William & Mary": 3.6,
        "Virginia Tech College of Arts and Sciences": 3.4,
        "Virginia Tech College of Engineering": 3.2,
        "Virginia Tech College of Agriculture & Life Sciences": 3.0,
        "George Mason University": 2.85,
        "Virginia Commonwealth University": 2.5,
        "Regis University": 2.0,
        "Shenandoah University": 2.5,
        "Old Dominion University": 2.5,
        "Mary Baldwin University": 2.5,
        "Regent University": 2.5,
        "Liberty University": 2.0,
        "James Madison University": 3.0,
        "Christopher Newport University": 3.5,
        "Longwood University": 2.5,
        "University of Mary Washington": 3.25,
        "University of Lynchburg": 2.0,
        "Radford University": 2.8,
        "Ferrum University": 2.0,
        "Bluefield University": 2.5,
        "Virginia State University": 2.0,
        "Troy University": 2.0,
        "Norfolk State University": 2.0,
        "ECPI University": 2.5,
        "Strayer University": 2.0,
        "Hollins University": 2.5,
        "Randolph University": 2.7,
        "Emory and Henry College": 2.5,
        "University of Virginia's College at Wise": 2.5,
        "Virginia Wesleyan College": 2.5,
        "Virginia Union University": 2.0
    }
    colleges = dict(sorted(colleges.items()))
    accepted_colleges = []
    for key, value in colleges.items():
        if gpa >= value:
            accepted_colleges.append(key)
    accepted_colleges = ", ".join(accepted_colleges)
    if len(accepted_colleges) > 0:
        print(f"You have the GPA required to attend:\n{accepted_colleges}")
    else:
        print("Your GPA is too low to attend any college in the state of Virginia.")


if __name__ == "__main__":
    main()
```