CT108-3-1-PYP

INDIVIDUAL ASSIGNMENT

INTAKE CODE: APU1F2009CS

Module Lecturer: Tanveer Khaleel Shaikh

Name: Ryan Martin

TP Number: TP058091

Table of Contents

Introduction

Real Champions Sport Academy is one of the fastest growing sport centers in Malaysia. They have branches in multiple cities in Malaysia, and employ tens of coaches to teach in a variety of sports, including swimming, badminton, volleyball, etc. All Real Champion's data, including their coach records, student records, and class schedules are managed by an admin staff from their human resources department.

Currently, the record filing system is done manually by the admin. It is done by categorizing coaches and students by their respective sport center. The records are sorted alphabetically by their name. The main problem with this system, is that searching for could be difficult as the record can be misplaced in another sport center, or it can also be in the placed in the wrong alphabet sequence. The admin then decided to computerize their record filing system. In this report, I will explain my solution for their problem.

Assumptions

For this task, I made several assumptions about the system. The first thing I assumed is that all their sport centers open at 08:00 MYT and closes at 17:00 MYT. There will be 10 different types of sports, they are: swimming, badminton, football, archery, gymnastics, volleyball, basketball, cricket, tennis, and table tennis. Each coach is only allowed to teach one specific sport and in a specific sport center only. All coaches are paid an hourly rate based on their respective sport they're specializing in. The hourly rate is between RM 100 to RM 500 inclusive. Also, coaches can be given ratings by students. The ratings can only be integer values 1 to 5.

Design – Pseudocode

1. Program entrypoint

```
BEGIN
 CALL init()
 CALL main()
END
```

    1. init() function

```
DEF init():
BEGIN
 DECLARE files, exists, coach, sport, student, sport_centre, schedule, feedback, rating
 files = ['coach.txt', 'student.txt', 'schedule.txt', 'sport.txt',
       'sport-centre.txt', 'feedback.txt', 'rating.txt']
 LOOP i FROM 0 TO len(files) STEP BY 1:
  exists = CALL os.path.exists("data/" + files[i])
  IF (exists == False):
   OPEN ("data/" + records[i]) MODE "w"
   CLOSE file
  ENDIF
 ENDLOOP
 coach = OPEN "data/coach.txt" MODE "r+"
  IF (READ coach == ""):
   WRITE
"coach_id,coach_name,coach_date_joined,coach_date_terminated,coach_phone,coach_address,coach_sport_id,coach_sport_centr
e_id,coach_rating\n"
  ENDIF
```

```
 CLOSE coach
 student = OPEN "data/student.txt" MODE "r+"
  IF (READ student == ""):
    WRITE "student_id,student_name,student_date_joined,student_email,student_password,student_phone,student_address\n"
  ENDIF
 CLOSE student
 schedule = OPEN "data/schedule.txt" MODE "r+"
  IF (READ schedule == ""):
    WRITE "schedule_id,schedule_date,schedule_time,schedule_duration,schedule_coach_id,schedule_booked_ids\n"
  ENDIF
 CLOSE schedule
 sport = OPEN "data/sport.txt" MODE "r+"
  IF (read SPORT == ""):
    WRITE "sport_id,sport_name,sport_hourly_fee\n"
  ENDIF
 CLOSE sport
 sport_centre = OPEN "data/sport-centre.txt" MODE "r+"
  IF (READ coach == ""):
    WRITE "sport_centre_id,sport_centre_name,sport_centre_location\n"
  ENDIF
 CLOSE sport_centre
 rating = OPEN "data/rating.txt" MODE "r+"
  IF (READ rating == ""):
    WRITE "rating_id,rating_coach_id,rating_student_id,rating_value"
  ENDIF
 CLOSE rating
 feedback = OPEN "data/feedback.txt" MODE "r+"
  IF (READ coach == ""):
    WRITE "coach_id,feedback"
  ENDIF
 CLOSE feedback
END
```

2. main() function

```
DEF main():
BEGIN
 DECLARE inp, success
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "REAL CHAMPIONS SPORT CENTRE\n1. Admin\n2. Student\n3. Quit"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3"],
                        "invalid choice, enter a valid choice")
  IF (inp == "1"):
    success = CALL admin_login()
    IF (success == True):
      CALL admin_menu()
    ENDIF
  ELIF (inp == "2"):
    CALL unregistered_student_menu()
  ELSE:
    DISPLAY "quitting..."
    RETURN
 ENDWHILE
END
```

3. admin_login() function:

```
DEF admin_login():
BEGIN
 DECLARE inp, max_tries
 CALL os.system("clear | cls")
 DISPLAY "ADMIN LOGIN"
```

```
 DISPLAY "enter password: "
 READ inp
 max_tries = 4
 WHILE (inp != "password" AND max_tries > 0):
  DISPLAY "wrong password, you have ", max_tries, " tries left"
  DISPLAY "enter password: "
  READ inp
  max_tries -= 1
 ENDWHILE
 IF (max_tries == 0):
  DISPLAY "too many failed attempts, press enter to continue: "
  RETURN False
 ENDIF
 RETURN True
END
```

4. admin_menu() function:

```
DEF admin_menu():
BEGIN
 DECLARE inp
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "ADMIN MENU\n1. Add Records\n2. Display Records\n3. Search Specific Records"
  DISPLAY "4. Sort Records\n5. Modify Records\n6. Quit"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5", "6"],
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   CALL admin_add_records()
  ELIF (inp == "2"):
   CALL admin_display_records()
  ELIF (inp == "3"):
   CALL admin_search_records()
  ELIF (inp == "4"):
   CALL admin_sort_records()
  ELIF (inp == "5"):
   CALL admin_modify_records()
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

5. admin_add_records() function:

```
DEF admin_add_records():
BEGIN
 DECLARE inp
 WHILE
  CALL os.system("clear | cls")
  DISPLAY "ADD RECORDS"
  DISPLAY "1. Add Coach Record\n2. Add Schedule Record\n3. Add Sport Record"
  DISPLAY "4. Add Sport Schedule Record\n5. Return to Admin Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"],
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   CALL add_coach_record()
  ELIF (inp == "2"):
   CALL add_schedule_record()
```

```
  ELIF (inp == "3"):
   CALL add_sport_record()
  ELIF (inp == "4"):
   CALL add_sport_centre_record()
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

6. add_coach_records() function:

```
DEF add_coach_record():
BEGIN
 DECLARE coach_records, sport_records, sport_centre_records
 DECLARE name, date_joined, date_terminated, phone, address, sport, sport_list
 DECLARE sport_centre, sport_centre_list, coach_id, sport_id, sport_centre_id
 DECLARE coach_rating, record
 coach_records = CALL read_csv("data/coach.txt")
 sport_records = CALL read_csv("data/sport.txt")
 sport_centre_records = CALL read_csv("data/sport-centre.txt")
 DISPLAY "enter coach name"
 READ name
 name = CALL validate_empty_input(name, "coach name")
 DISPLAY "enter date joined (in DD-MM-YYYY format), leave blank for current date: "
 READ date_joined
 date_joined = CALL validate_date_input(date_joined)
 DISPLAY "enter date joined (leave blank to skip): "
 READ date_terminated
 date_terminated = CALL validate_date_input(date_terminated, empty=True)
 DISPLAY "enter coach phone number: "
 READ phone
 phone = CALL validate_number_input(phone, "phone number")
 DISPLAY "enter coach address: "
 READ address
 address = CALL validaate_empty_input(address, "coach address")
 DISPLAY "enter sport name: "
 READ sport
 sport_list = CALL select_column_from_table("sport_name", sport_records)
 sport = CALL validate_choice_input(sport, sport_list, "invalid sport, enter a valid choice: ")
 DISPLAY "enter sport centre name: "
 READ sport_centre
 sport_centre_list = CALL select_column_from_table("sport_centre_name", sport_centre_records)
 sport_centre = CALL validate_choice_input(sport_centre, sport_centre_list,
                          "invalid sport centre, enter a valid choice: ")
 coach_id = CALL select_column_from_table("coach_id", coach_records)
 coach_id = CALL len(coach_id)
 sport_id = CALL select_column_from_table_where("sport_id", "sport_name", sport, sport_records)
 sport_id = sport_id[0]
 sport_centre_id = CALL select_column_from_table_where("sport_centre_id", "sport_centre_name", sport_centre,
sport_centre_records)
 sport_centre_id = sport_centre_id[0]
 coach_rating = "0"
 record = [str(coach_id), name, date_joined, date_terminated, phone,
       address, sport_id, sport_centre_id, coach_rating]
 CALL write_record(record, "data/coach.txt")
END
```

7. add_schedule_record() function:

```
DEF add_schedule_records():
BEGIN
 DECLARE coach_records, schedule_records, date, time, duration, okay
```

```
 DECLARE coach_id, coach_id_list, schedule_id, record
 coach_records = CALL read_csv("data/coach.txt")
 schedule_records = CALL read_csv("data/schedule.txt")
 DISPLAY "enter date joined (in DD-MM-YYYY format), leave blank for current date: "
 READ date
 date = CALL validate_date_input(date)
 DISPLAY "enter class time (24-hour format HH:MM): "
 READ time
 time = CALL validate_time_input(time)
 DISPLAY "enter duration (between 1-9 hours)"
 READ duration
 duration = CALL validate_duration_input(duration, time)
 DISPLAY "enter coach id: "
 READ coach_id
 coach_id_list = CALL select_column_from_table("coach_id", coach_records)
 coach_id = CALL validate_choice_input(coach_id, coach_id_list,
                          "invalid coach id, enter a valid choice: ")
 okay = CALL validate_schedule(date, time, duration, coach_id)
 IF (okay == True):
  schedule_id = CALL select_column_from_table("schedule_id", schedule_records)
  schedule_id = CALL len(schedule_id)
  record = [str(schedule_id), date, time, duration, coach_id, ""]
  CALL write_record(record, "data/schedule.txt")
 ELSE:
  DISPLAY "coach is already booked! press enter to continue: "
  READ
 ENDIF
END
```

8. add_sport_record() function:

```
DEF add_sport_record():
BEGIN
 DECLARE sport_records, name, fee, sport_id, record
 sport_records = CALL read_csv("data/sport.txt")
 DISPLAY "enter sport name: "
 READ name
 name = CALL validate_empty_input(name, "sport name")
 DISPLAY "enter hourly fee: "
 READ fee
 fee = CALL validate_fee_input(fee)
 sport_id = CALL select_column_from_table("sport_id", sport_records)
 sport_id = CALL len(sport_id)
 record = [str(sport_id), name, fee]
 CALL write_record(record, "data/sport.txt")
END
```

9. add_sport_centre_record() function:

```
DEF add_sport_centre_record():
BEGIN
 DECLARE sport_centre_records, name, loation, sport_centre_id, record
 sport_centre_records = CALL read_csv("data/sport-centre.txt")
 DISPLAY "enter sport centre name: "
 READ name
 name = CALL validate_empty_input(name, "sport centre name")
 DISPLAY "enter location: "
 READ location
 location = CALL validate_empty_input(location)
 sport_centre_id = CALL select_column_from_table("sport_centre_id", sport_centre_records)
 sport_centre_id = CALL len(sport_centre_id)
 record = [str(sport_centre_id), name, location]
 CALL write_record(record, "data/sport-centre.txt")
```

```
END
```

10. admin_display_records() function:

```
DEF admin_display_record():
BEGIN
 DECLARE inp, record
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "DISPLAY RECORDS"
  DISPLAY "1. Display Coach Records\n2. Display Student Records"
  DISPLAY "3. Display Schedule Records\n4. Display Sport Records"
  DISPLAY "5. Display Sport Centre Records\n6. Return to Admin Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5", "6"]
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   record = CALL read_csv("data/coach.txt")
  ELIF (inp == "2"):
   record = CALL read_csv("data/student.txt")
  ELIF (inp == "3"):
   record = CALL read_csv("data/schedule.txt")
  ELIF (inp == "4"):
   record = CALL read_csv("data/sport.txt")
  ELIF (inp == "5"):
   record = CALL read_csv("data/sport-centre.txt")
  ELSE:
   RETURN
  ENDIF
  CALL pretty_print(record)
 ENDWHILE
END
```

11. admin_search_records() function:

```
DEF admin_search_record():
BEGIN
 DECLARE inp
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "SEARCH RECORDS"
  DISPLAY "1. Search for Coach Record\n2. Search for Student Record"
  DISPLAY "3. Search for Schedule Record\n4. Search for Sport Record"
  DISPLAY "5. Search for Sport Centre Record\n6. Return to Admin Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5", "6"]
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   CALL search_coach_records()
  ELIF (inp == "2"):
   CALL search_student_records()
  ELIF (inp == "3"):
   CALL search_schedule_records()
  ELIF (inp == "4"):
   CALL search_sport_records()
  ELIF (inp == "5"):
   CALL search_sport_centre_records()
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

12. search_coach_records() function:

```
DEF search_coach_records():
BEGIN
 DECLARE inp, coach_records, sport_records, sport_centre_records
 DECLARE coach_sport_records, columns, coach_sport_centre_records
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "SEARCH COACH RECORDS\nWhat do you want to search by ?"
  DISPLAY "1. Coach ID\n2. Name\n3. Phone Number\n4. Sport Name"
  DISPLAY "5. Sport Centre Name\n6. Rating\n7. Go Back"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5", "6", "7"]
                    "invalid choice, enter a valid choice: ")
  coach_records = CALL read_csv("data/coach.txt")
  sport_records = CALL read_csv("data/sport.txt")
  sport_centre_records = CALL read_csv("data/sport-centre.txt")
  IF (inp == "1"):
   CALL search_records_by("coach_id", "coach id", coach_records)
  ELIF (inp == "2"):
   CALL search_records_by("coach_name", "coach name", coach_records)
  ELIF (inp == "3"):
   CALL search_records_by("coach_phone", "coach phone number", coach_records)
  ELIF (inp == "4"):
   coach_sport_records = CALL left_join_records_on(coach_records, "coach_sport_id",
                                 sport_records, "sport_id")
   columns = ["coach_id", "coach_name", "coach_date_joined",
         "coach_date_terminated", "coach_phone", "coach_address",
         "coach_soprt_id", "coach_sport_centre_id", "coach_rating",
         "sport_name"]
   coach_sport_reecords = CALL select_columns_from_table(columns,
                                                coach_sport_records)
   CALL search_records_by("sport_name", "coach sport name", coach_sport_records)
  ELIF (inp == "5"):
   coach_soprt_centre_records = CALL left_join_records_on(coach_records, "coach_sport_centre_id")
   columns = ["coach_id", "coach_name", "coach_date_joined",
         "coach_date_terminated", "coach_phone", "coach_address",
         "coach_soprt_id", "coach_sport_centre_id", "coach_rating",
         "sport_centre_name"]
   coach_sport_centre_records = CALL select_columns_from_table(columns, coach_sport_centre_records)
   CALL search_records_by("sport_centre_name", "coach sport centre",
coach_sport_centre_records)
  ELIF (inp == "6"):
   CALL search_records_by("coach_rating", "coach rating", coach_records)
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

13. search_student_records() function:

```
DEF search_student_records():
BEGIN
 DECLARE inp, student_records
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "SEARCH STUDENT RECORDS\nwhat do you want to search by ?"
  DISPLAY "1. Student ID\n2. Name\n3. Phone Number\n4. Email\n5. Go Back"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"]
```

```
                    "invalid choice, enter a valid choice: ")
  student_records = CALL read_csv("data/student.txt")
  IF (inp == "1"):
   CALL search_records_by("student_id", "student id", student_records)
  ELIF (inp == "2"):
   CALL search_records_by("student_name", "student name", student_records)
  ELIF (inp == "3"):
   CALL search_records_by("student_phone", "student phone", student_records)
  ELIF (inp == "4"):
   CALL search_records_by("student_email", "student email", student_records)
  ELSE:
   RETURN
  ENDIF
  DISPLAY "press enter to go back: "
  READ
 ENDWHILE
END
```

14. search_schedule_records() function:

```
DEF search_schedule_records():
BEGIN
 DECLARE inp, schedule_records, query, seconds, time_index, duration_index
 DECLARE, records, lower_bound, upper_bound
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY " SEARCH SCHEDULE RECORDS\nwhat do you want to search by ?"
  DISPLAY "1. Date\n2. Time\n3. Duration\n4. Coach ID\n5. Go Back"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"]
                    "invalid choice, enter a valid choice: ")
  schedule_records = CALL read_csv("data/schedule.txt")
  IF (inp == "1"):
   CALL search_records_by("schedule_date", "schedule date", schedule_records)
  ELIF (inp == "2"):
   DISPLAY "enter class time (HH:MM 25-hour format): "
   READ query
   query = CALL validate_time_input(query)
   seconds = CALL hhmm_to_seconds(query)
   time_index = CALL get_column_index("schedule_time", schedule_records)
   duration_index = CALL get_column_index("schedule_duration", schedule_records)
   records = [schedule_records[0]]
   LOOP i FROM 0 to len(schedule_records[1:]) STEP BY 1:
    lower_bound = CALL hhmm_to_seconds(schedule_records[i][time_index])
    upper_bound = lower_bound + int(schedule_records[i][duration_index]) * 3600
    IF (seconds >= lower_bound AND seconds < upper_bound):
     records.append(schedule_records[i])
    ENDIF
   IF (len(records) > 1):
    CALL pretty_print(records, interactive=False)
   ELSE:
    DISPLAY "there is no record of class with time of ", query
   ENDIF
  ELIF (inp == "3"):
   CALL search_records_by("schedule_duration", "schedule duration",                    schedule_records)
  ELIF (inp == "4"):
   CALL search_records_by("schedule_coach_id", "coach id", schedule_records)
  ELSE:
   RETURN
  ENDIF
  DISPLAY "press enter to go back: "
  READ
```

```
 ENDWHILE
END
```

15. search_sport_records() function:

```
DEF search_sport_records():
BEGIN
 DECLARE inp, sport_records
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "SEARCH SPORT RECORDS\nwhat do you want to search by"
  DISPLAY "1. Sport ID\n2. Sport Name\n3. Hourly Fee\n4. Go Back"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4"]
                    "invalid choice, enter a valid choice: ")
  sport_records = CALL read_csv("data/sport.txt")
  IF (inp == "1"):
   CALL search_records_by("sport_id", "sport id", sport_records)
  ELIF (inp == "2"):
   CALL search_records_by("sport_name", "sport name", sport_records)
  ELIF (inp == "3"):
   CALL search_records_by("sport_hourly_fee", "hourly fee", sport_records)
  ELSE:
   RETURN
  ENDIF
  DISPLAY "press enter to continue: "
  READ
 ENDWHILE
END
```

16. search_sport_centre_records() function:

```
DEF search_sport_centre_records():
BEGIN
 DECLARE inp, sport_records
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "SEARCH SPORT CENTRE RECORDS\nwhat do you want to search by"
  DISPLAY "1. Sport Centre ID\n2. Sport Centre Name"
  DISPLAY "3. Location\n4. Go Back"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4"]
                    "invalid choice, enter a valid choice: ")
  sport_centre_records = CALL read_csv("data/sport-centre.txt")
  IF (inp == "1"):
   CALL search_records_by("sport_centre_id", "sport centre id", sport_centre_records)
  ELIF (inp == "2"):
   CALL search_records_by("sport_centre_name", "sport centre name", sport_centre_records)
  ELIF (inp == "3"):
   CALL search_records_by("sport_centre_location", "location", sport_centre_records)
  ELSE:
   RETURN
  ENDIF
  DISPLAY "press enter to continue: "
  READ
 ENDWHILE
END
```

17. admin_sort_records() function:

```
DEF admin_sort_records():
```

```
BEGIN
 DECLARE inp, asc
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "DISPLAY SORTED RECORDS"
  DISPLAY "1. Sort Coach by Coach ID\n2. Sort Coach by Name"
  DISPLAY "3. Sort Coach by Hourly Fee\n4. Sort Sport by Hourly Fee\n5. Return to Admin Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"]
                        "invalid choice, enter a valid choice: ")
  IF (inp == "5"):
   RETURN
  ELSE:
   DISPLAY "display records in ascending order? [Y/n]: "
   READ asc
   asc = CALL validate_choice_input(asc, ["y", "n"],
                         "invalid choice, enter a valid choice: ")
   IF (inp == "1"):
    CALL sort_coach_by_id(asc)
   ELIF (inp == "2"):
    CALL sort_coach_by_name(asc)
   ELIF (inp == "3"):
    CALL sort_coach_by_fee(asc)
   ELSE:
    CALL sort_sport_by_fee(asc)
   ENDIF
  ENDIF
 ENDWHILE
END
```

18. sort_coach_by_id() function:

```
DEF sort_coach_by_id():
BEGIN
 DECLARE coach_records, sorted_records
 coach_records = CALL read_csv("data/coach.txt")
 sorted_records = CALL bubble_sort_records("coach_id", coach_records)
 IF (asc == "y"):
  CALL pretty_print(sorted_records)
 ELSE:
  sorted_records = CALL reversed_records(sorted_records)
  CALL pretty_print(sorted_records)
 ENDIF
END
```

19. sort_coach_by_name() function:

```
DEF sort_coach_by_name():
BEGIN
 DECLARE coach_records, sorted_records
 coach_records = CALL read_csv("data/coach.txt")
 sorted_records = CALL bubble_sort_records("coach_name", coach_records)
 IF (asc == "y"):
  CALL pretty_print(sorted_records)
 ELSE:
  sorted_records = CALL reversed_records(sorted_records)
  CALL pretty_print(sorted_records)
 ENDIF
END
```

20. sort_coach_by_fee() function:

```
DEF sort_coach_by_fee():
BEGIN
 DECLARE coach_records, sport_records, merged_records, sorted_records, columns
 coach_records = CALL read_csv("data/coach.txt")
 sport_records = CALL read_csv("data/sport.txt")
 merged_records = CALL left_join_records_on(coach_records, "coach_sport_id",
                                sport_records, "sport_id")
 sorted_records = CALL bubble_sort_records("sport_hourly_fee", merged_records, )
 columns = ["coach_id", "coach_name", "coach_date_joined", "coach_date_terminated",          "sport_name",
"coach_sport_centre_id", "coach_rating",                                                    "sport_hourly_fee"]
 sorted_records = CALL select_columns_from_table(columns, sorted_records)
 IF (asc == "y"):
  CALL pretty_print(sorted_records)
 ELSE
  sorted_records = CALL reversed_records(sorted_records)
  CALL pretty_print(sorted_records)
 ENDIF
END
```

21. sort_sport_by_fee() function:

```
DEF sort_sport_by_fee():
BEGIN
 DECLARE sport_records, sorted_records
 sport_records = CALL read_csv("data/sport.txt")
 sorted_records = CALL bubble_sort_records("sport_hourly_fee", sport_records)
 IF (asc == "y"):
  CALL pretty_print(sorted_records)
 ELSE:
  sorted_records = CALL reversed_records(sorted_records)
  CALL pretty_print(sorted_records)
 ENDIF
END
```

22. admin_modify_records() function:

```
DEF admin_modify_records():
BEGIN
 DECLARE inp
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "MODIFY RECORDS"
  DISPLAY "1. Modify Coach Records\n2. Modify Schedule Records"
  DISPLAY "3. Modify Sport Records\n4. Modify Sport Centre Records\n5. Return to Admin Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"]
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   CALL modify_coach_records()
  ELIF (inp == "2"):
   CALL modify_schedule_records()
  ELIF (inp == "3"):
   CALL modify_sport_records()
  ELIF (inp == "4"):
   CALL modify_sport_centre_records()
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

23. modify_coach_records() function:

```
DEF modify_coach_records():
BEGIN
 DECLARE coach_records, sport_records, sport_centre_records, coach_id
 DECLARE valid_ids, name, date_joined, date_terminated, phone, address,
 DECLARE sport, sport_list, sport_centre, sport_centre_list, rating, record
 DECLARE new_records, okay
 coach_records = CALL read_csv("data/coach.txt")
 sport_records = CALL read_csv("data/sport.txt")
 sport_centre_records = CALL read_csv("data/sport-centre.txt")
 DISPLAY "enter coach id to modify: "
 READ coach_id
 valid_ids = CALL select_column_from_table("coach_id", coach_records)
 coach_id = CALL validate_choice_input(coach_id, valid_ids,
                        "invalid coach id, enter a valid id: ")
 DISPLAY "enter new coach name (leave blank to skip): "
 READ name
 DISPLAY "enter new date joined (leave blank to skip): "
 READ date_joined
 date_joined = CALL validate_date_input(date_joined, empty=True)
 DISPLAY "enter new date terminated (leave blank to skip): "
 READ date_terminated
 date_terminated = CALL validate_date_input(date_terminated, empty=True)
 DISPLAY "enter new coach phone number (leave blank to skip): "
 READ phone
 phone = CALL validate_number_input(phone, empty=True)
 DISPLAY "enter new coach address (leave blank to skip):"
 READ address
 DISPLAY "enter new sport id (leave blank to skip): "
 READ sport
 sport_list = CALL select_column_from_table("sport_id", sport_records)
 sport = CALL validate_choice_input(sport, sport_list,
                    "invalid sport id, enter a valid id", empty=True)
 DISPLAY "enter new sport centre id (leave blank to skip): "
 READ sport_centre
 sport_centre_list = CALL select_column_from_table("sport_centre_id",
                                    sport_centre_records)
 sport_centre = CALL validate_choice_input(sport_centre, sport_centre_list,
                        "invalid sport centre id, enter a valid id: ", empty=True)
 DISPLAY "enter new coach rating (leave blank to skip): "
 READ rating
 rating = CALL validate_choice_input(rating, ["1", "2", "3", "4", "5"]
                    "invalid choice, enter a valid choice: ", empty=True)
 record = [str(coach_id), name, date_joined, date_terminated, phone, address,
       sport, sport_centre, rating]
 new_records = CALL modify_record(record, coach_records)
 new_records = CALL select_row_from_table_where("coach_id", "coach id", new_records)
 CALL pretty_print(new_records, interactive=False)
 DISPLAY "enter y to change, n to cancel: "
 READ okay
 okay = validate_choice_input(okay, ["y", "n"],
                    "invalid choice, enter either y or n: ")
 IF (okay == "y"):
  CALL write_records(new_records, "data/coach.txt")
 ELSE:
  DISPLAY "cancelling... press enter to continue: "
  READ
 ENDIF
END
```

24. modify_schedule_records() function:

```
DEF modify_schedule_records():
BEGIN
```

```
 DECLARE schedule_records, coach_records, schedule_id, valid_ids, date, time,
 DECLARE duration, coach_id, coach_id_list, okay_schedule, record
 DECLARE new_records, okay
 schedule_records = CALL read_csv("data/schedule.txt")
 coach_records = CALL read_csv("data/coach.txt")
 DISPLAY "enter schedule id to modify: "
 READ schedule_id
 valid_ids = CALL select_column_from_table("schedule_id", schedule_records)
 schedule_id = CALL validate_choice_input(schedule_id, valid_ids,
                          "invalid schedule id, enter a valid id: ")
 DISPLAY "enter new class date (leave blank to skip): "
 READ date
 date = CALL validate_date_input(date, empty=True)
 DISPLAY "enter enw class time (leave blank to skip): "
 READ time
 time = CALL validate_time_input(time, empty=True)
 DISPLAY "enter new duration (between 1-9 hours): "
 READ duration
 duration = CALL validate_duration_input(duration, time, empty=True)
 DISPLAY "enter new coach id (leave blank to skip): "
 READ coach_id
 coach_id_list = CALL select_column_from_table("coach_id", coach_records)
 coach_id = CALL validate_choice_input(coach_id, coach_id_list,
                        "invalid coach id, enter valid coach id: ", empty=True)
 record = [str(schedule_id), date, time, duration, coach_id, '']
 old_records = select_row_from_table_where("schedule_id", schedule_id, schedule_records)[1]
 LOOP i FROM 0 TO len(record) STEP BY 1validate:
  IF (record[i] == ""):
   record[i] = old_records[i]
  ENDIF
 ENDLOOP
 okay_schedule = validate_schedule(record[0], record[1], record[2], record[3], record[4], empty=True)
 IF (okay_scheule == True):
  new_records = modify_record(record, schedule_records)
  new_records = CALL select_column_from_table_where("schedule_id", schedule_id, new_records)
  CALL pretty_print(new_records, interactive=False)
  DISPLAY "enter y to change, n to cancel: "
  READ okay
  okay = CALL validate_choice_input(okay, ["y", "n"],
                         "invalid choice, enter either y or n")
  IF (okay == "y"):
   CALL write_records(new_record, "data/schedule.txt")
  ELSE:
   DISPLAY "cancelling... press enter to continue: "
   READ
  ENDIF
 ELSE:
  DISPLAY "invalid schedule. press enter to try again: "
  READ
 ENDIF
END
```

25. modify_sport_records() function:

```
DEF modify_sport_records():
BEGIN
 DECLARE sport_records, sport_id, valid_ids, sport_id, name, fee, record
 DECLARE new_records, okay
 sport_records = CALL read_csv("data/sport.txt")
 DISPLAY "enter sport id to modify: "
 READ sport_id
 valid_ids = CALL select_column_from_table("sport_id", sport_records)
 sport_id = CALL validate_choice_input(sport_id, valid_ids,
```

```
                        "invalid sport id, enter a valid id: ")
 DISPLAY "enter new sport name (leave blank to skip): "
 READ name
 DISPLAY "enter new hourly fee (leave blank to skip): "
 READ fee
 fee = CALL validate_fee_input(fee, empty=True)
 record = [str(sport_id), name, fee]
 new_records = modify_record(record, sport_records)
 new_records = CALL select_row_from_table_where("sport_id", sport_id, new_records)
 CALL pretty_print(new_records, interactive=False)
 DISPLAY "enter y to change, n to cancel: "
 READ okay
 okay = CALL validate_choice_input(okay, ["y", "n"],
                        "invalid choice, enter either y or n")
 IF (okay == "y"):
  CALL write_records(new_records, "data/sport.txt")
 ELSE:
  DISPLAY "cancelling... press enter to continue: "
  READ
 ENDIF
END
```

26. modify_sport_centre_records() function:

```
DEF modify_sport_centre_records():
BEGIN
 DECLARE sport_centre_records, sport_centre_id, valid_ids, sport_centre_id
 DECLARE name, location, record, new_records, okay
 sport_centre_records = CALL read_csv('data/sport-centre.txt')
 DISPLAY "enter sport centre id to modify: "
 READ sport_centre_id
 valid_ids = CALL select_column_from_table("sport_centre_id", sport_centre_records)
 sport_centre_id = CALL validate_choice_input(sport_centre_id, valid_ids,
                        "invalid sport centre id, enter a valid id: ")
 DISPLAY "enter new sport centre name (leave blank to skip): "
 READ name
 DISPLAY "enter new location (leave blank to skip): "
 READ location
 record = [str(sport_centre_id), name, location]
 new_records = modify_record(record, sport_centre_records)
 new_records = CALL select_row_from_table_where("sport_centre_id", sport_centre_id, new_records)
 CALL pretty_print(new_records, interactive=False)
 DISPLAY "enter y to change, n to cancel: "
 READ okay
 okay = CALL validate_choice_input(okay, ["y", "n"],
                        "invalid choice, enter either y or n")
 IF (okay == "y"):
  CALL write_records(new_records, "data/sport-centre.txt")
 ELSE:
  DISPLAY "cancelling... press enter to continue: "
  READ
 ENDIF
END
```

27. unregistered_student_menu() function:

```
DEF unregistered_student_menu():
BEGIN
 DECLARE inp, student_id
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "STUDENT MENU"
  DISPLAY "1. View Details\n2. Login\n3. Register\n4. Quit"
```

```
  DISPLAY "enter choice: "
  READ inp
  inp = validate_choice_input(inp, ["1", "2", "3", "4"],
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   CALL unregistered_student_view_details()
  ELIF (inp == "2"):
   student_id = CALL student_login()
   IF (student_id != "0"):
    CALL registered_student_menu(student_id)
    RETURN
   ENDIF
  ELIF (inp == "3"):
   CALL student_register()
  ELSE:
   RETURN
  ENDIF
 ENDWHILE
END
```

28. unregistered_student_view_details() function:

```
DEF unregistered_student_view_details():
BEGIN
 DECLARE inp, records
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "VIEW DETAILS"
  DISPLAY "1. View Sport Details\n2. View Sport Schedule\n3. Return to Student Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3"],
                    "invalid choice, enter a valid choice: ")
  IF (inp == "1"):
   records = CALL read_csv("data/sport.txt")
  ELIF
   records = CALL get_pretty_schedule()
  ELSE
   RETURN
  ENDIF
  CALL pretty_print(records)
 ENDWHILE
END
```

29. student_login() function:

```
DEF student_login():
BEGIN
 DECLARE email, password, max_tries, student_records, student_id
 CALL os.system("clear | cls")
 DISPLAY "STUDENT LOGIN"
 DISPLAY "enter your email:"
 READ email
 email = CALL validate_empty_input(email, "email")
 DISPLAY "enter your password"
 READ password
 password = CALL validate_empty_input(password, "password")
 max_tries = 4
 WHILE True:
  DISPLAY "invalid credentials, you have ", max_tries, " tries left"
  DISPLAY "enter your email:"
  READ email
  email = CALL validate_empty_input(email, "email")
```

```
  DISPLAY "enter your password"
  READ password
  password = CALL validate_empty_input(password, "password")
  max_tries -= 1
 ENDWHILE
 IF (max_tries == 0):
  DISPLAY "too many failed attempts, returning..."
  RETURN 0
 ELSE:
  student_records = CALL read_csv("data/student.txt")
  student_id = CALL select_column_from_table_where("student_id", "student_email",
                              email, student_records)
  RETURN student_id[0]
 ENDIF
END
```

30. student_register() function:

```
DEF student_register():
BEGIN
 DECLARE name, email, password, hashed_password, phone, address, student_id
 DECLARE date_joined, record, student_records
 CALL os.system("clear | cls")
 student_records = CALL read_csv("data/student.txt")
 DISPLAY "STUDENT REGISTER"
 DISPLAY "enter your name: "
 READ name
 name = CALL validate_empty_input(name, "name")
 DISPLAY "enter your email address: "
 READ email
 email = CALL validate_empty_input(email, "email")
 DISPLAY "enter your password (minimum 8 characters): "
 READ password
 password = CALL validate_password(password)
 hashed_password = CALL hash_password(password)
 DISPLAY "enter your phone number (leave blank to skip): "
 READ phone
 phone = CALL validate_number_input(phone, empty=True)
 DISPLAY "enter your addrss (leave blank to skip): "
 READ address
 student_id = CALL select_column_from_table("student_id", student_records)
 student_id = CALL len(student_id)
 date_joined = CALL get_date()
 record = [str(student_id), name, date_joined, email, hashed_password, phone, address]
 CALL write_record(record, "data/student.txt")
END
```

31. registered_student_menu() function:

```
DEF registered_student_menu(student_id):
BEGIN
 DECLARE inp
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "STUDENT MENU"
  DISPLAY "1. View Details\n2. Modify Self Record\3. Provide Coach Feedback"
  DISPLAY "4. Manage Schedule\n5. Quit"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5"],
                        "invalid choice, enter a valid choice")
  IF (inp == "1"):
   CALL registered_student_view_details(student_id)
```

```
  ELIF (inp == "2"):
    CALL modify_student_records(student_id)
  ELIF (inp == "3"):
    CALL registered_student_give_feedback(student_id)
  ELIF (inp == "4"):
    CALL registered_student_manage_schedule(student_id)
  ELSE:
    RETURN
  ENDIF
 ENDWHILE
END
```

32. registered_student_view_details() function:

```
DEF registered_student_view_details(student_id):
BEGIN
 DECLARE inp, records, student_records, coach_records, sport_records, ids
 DECLARE sport_centre_records, compound_records, columns, old_records, index
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "VIEW DETAILS"
  DISPLAY "1. View Sport Details\n2. View Profile\n3. View Coach Details"
  DISPLAY "4. View Sport Schedule\n5. View Booked Classes\n6. Return to Student Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3", "4", "5", "6"],
                    "invalid choice, enter a valid choice")
  IF (inp == "1"):
    records = CALL read_csv("data/sport.txt")
  ELIF (inp == "2"):
    student_records = CALL read_csv("data/student.txt")
    records = CALL select_row_from_table_where("student_id", student_id,
                      student_records)
  ELIF (inp == "3"):
    coach_records = CALL read_csv("data/coach.txt")
    sport_records = CALL read_csv("data/sport.txt")
    sport_centre_records = CALL read_csv("data/sport-centre.txt")
    compound_records = CALL left_join_records_on(coach_records, "coach_sport_id", sport_records, "sport_id")
    compound_records = CALL left_join_records_on(compound_records, "coach_sport_centre_id", sport_records, "sport_id")
    columns = ["coach_id", "coach_name", "coach_phone", "sport_name",
          "sport_centre_name", "coach_rating"]
    records = CALL select_columns_from_table(columns, compound_records)
  ELIF (inp == "4"):
    records = CALL get_pretty_schedule()
  ELIF (inp == "5"):
    old_records = CALL get_pretty_schedule()
    index = CALL get_column_index("schedule_booked_ids", old_records)
    records = [old_records[0][:-1]]
    LOOP i FROM 1 TO len(old_records) STEP BY 1:
      ids = old_records[i][index]
      LOOP j FROM 0 TO len(ids) STEP BY 1:
        IF (ids[j] == student_id):
          records.append(old_records[i])
        ENDIF
      ENDLOOP
    ENDLOOP
    IF (len(records) == 1):
      DISPLAY "you don't have any booked classes. press enter to continue: "
      READ
      RETURN
    ENDIF
  ELSE:
    RETURN
  ENDIF
```

```
  CALL pretty_print(records)
 ENDWHILE
END
```

33. modify_student_records() function:

```
DEF modify_student_records(student_id):
BEGIN
 DECLARE name, email, phone, address, password, student_records, record
 DECLARE new_records, okay
 DISPLAY "enter new name (leave blank to skip): "
 READ name
 DISPLAY "enter new email (leave blank to skip): "
 READ email
 DISPLAY "enter new phone (leave blank to skip): "
 READ phone
 phone = CALL validate_number_input(phone, empty=True)
 DISPLAY "enter new address (leave blank to skip): "
 READ address
 DISPLAY "enter new password (leave blank to skip): "
 READ password
 password = CALL validate_password(password, empty=True)
 student_records = CALL read_csv("data/student.txt")
 record = [student_id, name, "", email, password, phone, address]
 new_records = CALL modify_record(record, student_records)
 new_records = CALL select_row_from_table_where("student_id", student_id, new_records)
 CALL pretty_print(new_records, interactive=False)
 DISPALY "enter y to change, n to cancel: "
 READ okay
 okay = CALL validate_choice_input(okay, ["y", "n"],
                    "invalid choice, enter either y or n: ")
 IF (okay == "y"):
  CALL write_records(new_records, "data/student.txt")
 ELSE:
  DISPLAY "cancelling... press enter to continue"
  READ
 ENDIF
END
```

34. registered_student_give_feedback() function:

```
DEF registered_student_give_feedback(student_id):
BEGIN
 DECLARE inp, record, coach_records, rating_records, coach_id, coach_ids
 DECLARE rating, coach_id_index, student_id_index, value_index, old_rating
 DECLARE coach_rating, rating_count, rated, new_rating, new_rating_record
 DECLARE new_rating_records, rating id, feedback, record
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "RATE COACH AND GIVE FEEDBACK\n(feedbacks are anonymous)"
  DISPLAY "1. Rate Coach\n2. Provide Feedack\n3. Return to Student Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3"],
                    "invalid choice, enter a valid choice")
  IF (inp == "3"):
   RETURN
  ENDIF
  coach_records = CALL read_csv("data/coach.txt")
  rating_record = CALL read_csv("data/rating.txt")
  CALL pretty_print(record, interactive=False)
  DISPLAY "enter coach id: "
  READ coach_id
```

```
    coach_ids = CALL select_column_from_table("coach_id", coach_records)
    coach_id = CALL validate_choice_input(coach_id, coach_ids,
                                "invalid coach id, enter a valid id: ")
   IF (inp == "1"):
    DISPLAY "enter coach rating (from 1-5 points): "
    READ rating
    rating = CALL validate_choice_input(rating, ["1", "2", "3", "4", "5"],
                                "invalid rating, enter a valid rating: ")
    coach_id_index = CALL get_column_index("rating_coach_id", rating_records)
    student_id_index = CALL get_column_index("rating_student_id", rating_records)
    value_index = CALL get_column_index("rating_value", rating_records)
    old_rating = 0.0
    LOOP i FROM 0 TO len(rating_records) STEP BY 1:
     IF (rating_records[i][coach_id_index] == coach_id AND
        rating_records[i][student_id_index] == student_id):
       old_rating = rating_records[value_index]
       BREAK
     ENDIF
    ENDLOOP
    coach_rating = CALL select_column_from_table_where("coach_rating", "coach_id", coach_id, coach_records)
    coach_rating = coach_rating[0]
    rating_count = CALL select_column_from_table_where("rating_id", "rating_coach_id", coach_id, rating_records)
    rating_count = CALL len(rating_count)
    rated = CALL validate_rating(coach_id, student_id, rating_records)
    new_rating = CALL get_new_rating(rating, old_rating,
                                                        coach_rating, rating_count, rated)
    IF (rated == True):
     LOOP i FROM 0 TO len(rating_records) STEP BY 1:
      IF (rating_records[i][coach_id_index] == coach_id AND
         rating_records[i][student_id_index] == student_id):
        rating_id = rating_records[i][0]
      ENDIF
     ENDLOOP
     new_rating_record = [rating_id, "", "", rating]
     new_rating_records = CALL modify_record(new_rating_record, rating_records)
     CALL write_records(new_rating_records, "data/rating.txt", interactive=False)
    ELSE:
     rating_id = CALL select_column_from_table("rating_id", rating_records)
     rating_id = CALL len(rating_id)
     new_rating_record = [str(rating_id), coach_id, student_id, rating]
     CALL write_record(new_rating_record, "data/rating.txt", interactive=False)
     new_coach_record = [coach_id, "", "", "", "", "", "", "", new_rating]
     new_coach_records = CALL modify_record(new_coach_record, coach_records)
     CALL write_recorsd(new_coach_records, "data/coach.txt")
    ENDIF
   ELSE:
    DISPLAY "enter your feedback: "
    READ feedback
    IF (feedback == ""):
     DISPLAY "empty feedback, cancelling... press enter to continue: "
     READ
    ELSE:
     record = [coach_id, feedback]
     CALL write_record(record, "data/feedback.txt")
    ENDIF
  ENDWHILE
END
```

35. registered_student_manage_schedule() function:

```
DEF registerd_student_manage_schedule(student_id):
BEGIN
 DECLARE inp, old_records, index, schedule, ids, schedule_records
```

```
 DECLARE schedule_id, valid_ids, in_record, booked
 WHILE True:
  CALL os.system("clear | cls")
  DISPLAY "MANAGE SCHEDULE"
  DISPLAY "1. Book Class\n2. Cancel Class\n3. Return to Student Menu"
  DISPLAY "enter choice: "
  READ inp
  inp = CALL validate_choice_input(inp, ["1", "2", "3"],
                        "invalid choice, enter a valid choice")
  IF (inp == "3"):
   RETURN
  ENDIF
  old_records = CALL get_pretty_schedule(booked=True)
  index = CALL get_column_index("schedule_booked_ids", old_records)
  schedule = [old_records[0][:-1] + ["booked"]]
  LOOP i FROM 1 TO len(old_records) STEP BY 1:
   ids = old_records[i]
   in_record = False
   LOOP j FROM 0 TO len(ids) STEP BY 1:
    IF (ids[j] == student_id):
     in_record = True
     BREAK
    ENDIF
   ENDLOOP
   IF (in_record == True):
    schedule.append(old_records[i][:-1] + ["yes"])
   ELSE:
    schedule.append(old_records[i][:-1] + ["no"])
   ENDIF
  ENDLOOP
  CALL pretty_print(schedule, interactive=False)
  schedule_records = CALL read_csv("data/schedule.txt")
  DISPLAY "enter schedule id: "
  READ schedule_id
  valid_ids = CALL select_column_from_table("schedule_id", schedule_records)
  schedule_id = CALL validate_choice_input(schedule_id, valid_ids,
                          "invalid id, enter a valid id: ")
  booked = CALL is_booked(schedule_id, schedule)
  IF (inp == "1"):
   IF (booked == True):
    DISPLAY "class is already booked. press enter to continue: "
    READ
   ELSE:
    CALL change_schedule("book", schedule_id, schedule_records, student_id)
   ENDIF
  ELSE:
   IF (booked == False):
    DISPLAY "you haven't booked that class. press enter to continue: "
    READ
   ELSE:
    CALL change_schedule("cancel", schedule_id, schedule_records, student_id)
   ENDIF
  ENDIF
 ENDWHILE
END
```

36. validate_empty_input() function:

```
DEF validate_empty_input(inp, value):
BEGIN
 WHILE inp == "":
  DISPLAY value, " must not be empty, enter " value, ": "
  READ inp
```

```
 ENDWHILE
 RETURN inp
END
```

37. validate_choice_input() function:

```
DEF validate_choice_input(inp, valid_choices, msg, empty=False):
BEGIN
 IF (empty == True AND inp == ""):
  RETURN ""
 ENDIF
 WHILE True:
  LOOP i FROM 0 TO len(valid_choices) STEP BY 1:
   IF (valid_choices[i] == inp):
    RETURN inp
   ENDIF
   DISPLAY msg
   READ inp
  ENDLOOP
 ENDWHILE
END
```

38. validate_number_input() function:

```
DEF validate_number_input(inp, empty=False)
BEGIN
 WHILE True:
  IF (empty == True AND inp == ""):
   RETURN ""
  ENDIF
  IF (inp.isdigit() == True):
   RETURN inp
  ELSE:
   DISPLAY "invalid value, please enter a number: "
   READ inp
  ENDIF
 ENDWHILE
END
```

39. Validate_date_input() function:

```
DEF validate_date_input(inp, empty=False):
BEGIN
 DECLARE current_date, dup, day, month, year, isday, ismonth, isyear
 WHILE True:
  IF (empty == True AND inp == ""):
   RETURN ""
  ELIF (inp == ""):
   current_date = CALL get_date()
   RETURN current_date
  ELSE:
   IF (len(inp) == 10 AND inp[2] == "-", and inp[5] == "-"):
    dup = inp
    CALL split on dup
    day, month, year = dup[0], dup[1], dup[2]
    isday = CALL isdigit(day)
    ismonth = CALL isdigit(month)
    isyear = CALL isdigit(year)
    IF (isday == True AND ismonth == True AND isyear == True):
     IF ((day >= 1 AND day <= 31) AND (month >= 1 AND month <= 12)):
      RETURN inp
     ENDIF
    ENDIF
```

```
    ENDIF
    DISPLAY "invalid date, enter date (in DD-MM-YYYY format), leave blank for current date"
    READ inp
   ENDIF
  ENDWHILE
END
```

40. validate_time_input() function:

```
DEF validate_time_input(inp, empty=False):
BEGIN
  DECLARE seconds, second_cond
  WHILE True:
   IF (empty == True AND inp == ""):
    RETURN ""
   ENDIF
   second_cond = (CALL isdigit on inp[:2]) AND (CALL isdigit on inp[3:])
   IF ((len(inp) == 5 AND inp[2] == ":") AND (second_cond == True)):
    IF ((hour >= 0 AND hour <= 23) AND (min >= 0 and min <= 60)):
     seconds = hour * 3600 + min * 60
     IF (seconds >= (8 * 3600) AND seconds <= (16 * 3600)):
      RETURN inp
     ELSE
      DISPLAY "invalid time, please enter time betweeen 08:00 and 16:00"
      READ inp
     ENDIF
    ELSE
     DISPLAY "invalid time, enter time in 24-hour format (HH:MM)"
     READ inp
    ENDIF
   ELSE:
    DISPLAY "invalid time, enter time in 24-hour format (HH:MM)"
    READ inp
   ENDIF
  ENDWHILE
END
```

41. validate_feee_input() function:

```
DEF validate_fee_input(inp, empty=False):
BEGIN
  DECLARE digit
  WHILE
   IF (empty == True AND inp == ""):
    RETURN ""
   ENDIF
   digit = CALL isdigit on inp
   IF (digit == True):
    IF (inp >= 100 AND inp <= 500)):
     RETURN inp
    ELSE:
     DISPLAY "invalid fee, allowed values are between 100-500"
     READ inp
    ENDIF
   ELSE:
    DISPLAY "invalid fee, please enter a valid value"
    READ inp
   ENDIF
  ENDWHILE
END
```

42. Validate_duration_input() function:

```
DEF validate_duration_input(inp, time, empty=False)
BEGIN
 DECLARE digit
 WHILE True:
  IF (time == "" OR (empty == True AND inp == ""))
   RETURN ""
  ENDIF
  time = CALL hhmm_to_seconds(time)
  digit = CALL isdigit on inp
  IF (digit == True):
   IF ((inp >= 1 AND inp <= 9) AND (inp * 3600 + time <= (17 * 3600))):
    RETURN Inp
   ENDIF
  ENDIF
  DISPLAY "invalid duration, duration must be between 1-9 hours and class must not pass 17:00"
  DISPLAY "enter valid duration: "
  READ inp
 ENDWHILE
END
```

43. validate_schedule() function:

```
DEF validate_schedule(date, time, duration, coach_id, empty=False):
BEGIN
 DECLARE start_time, end_time, records, date_index, coach_id_index
 DECLARE time_index, duration_index, lower_bound, upper_bound, toskip
 records = CALL read_csv("data/schedule.txt")
 date_index = CALL get_column_index("schedule_date", records)
 coach_id_index = CALL get_column_index("schedule_coach_id", records)
 time_index = CALL get_column_index("schedule_time", records)
 duration_index = CALL get_column_index("schedule_duration", records)
 toskip = 0
 start_time = CALL hhmm_to_seconds(time)
 end_time = start_time + duration * 3600
 LOOP i FROM 0 TO len(schedule_records) STEP BY 1:
  IF (schedule_records[i][0] == schedule_id):
   toskip = schedule_records[i][0]
  ENDIF
 ENDLOOP
 LOOP i FROM 1 TO len(records) STEP BY 1:
  IF (date == records[i][date_index] AND coach_id == records[i][coach_index]):
   lower_bound = CALL hhmm_to_seconds(records[i][time_index])
   upper_bound = lower_bound + records[i][duration_index] * 3600
   IF (NOT ((start_time < lower_bound and end_time < lower_bound) \
     OR (start_time > upper_bound and end_time > upper_bound))):
    RETURN False
   ENDIF
  ENDIF
 ENDLOOP
 RETURN True
END
```

44. validate_password() function:

```
DEF validate_password(inp, empty=False):
BEGIN
 IF (empty == True AND inp == ""):
  RETURN ""
 ENDIF
 WHILE len(inp) < 8:
  DISPLAY "password must be atleast 8 characters!"
  DISPLAY "enter password: "
  READ inp
 ENDWHILE
```

```
 RETURN inp
END
```

45. validate_login() function:

```
DEF validate_login(email, password):
BEGIN
 DECLARE student_records, hashed
 student_records = CALL read_csv("data/student.txt")
 student_records = CALL select_columns_from_table(["student_email", "student_password"], student_records)
 hashed = CALL hash_password(password)
 LOOP i FROM 0 TO len(student_records) STEP BY 1:
  IF (student_records[i][0] == email AND hashed == student_records[i][1]):
   RETURN True
  ENDIF
 ENDLOOP
 RETURN False
END
```

46. validate_rating() function:

```
DEF validate_rating(coach_id, student_id, records):
BEGIN
 DECLARE coach_id_index, student_id_index, rating_id
 coach_id_index = CALL get_column_index("rating_coach_id", records)
 student_id_index = CALL get_column_index("rating_student_id", records)
 rating_id = CALL get_column_index("rating_value", records)
 LOOP i FROM 0 TO len(records) STEP BY 1:
  IF (records[i][coach_id_index] == coach_id AND
    records[i][student_id_index] == student_id):
   RETURN True
  ENDIF
 ENDLOOP
 RETURN False
END
```

47. read_csv() function:

```
DEF read_csv(filename):
BEGIN
 DECLARE records
 records = []
 OPEN filename MODE "r"
  READ lines INTO records
 CLOSE file
 LOOP i FROM 0 TO len(records) STEP BY 1:
  CALL split on records[i]
  LOOP j FROM 0 TO len(records[i]) SET BY 1:
   IF (records[i][j] == ""):
    records[i][j] = "-"
   ENDIF
  ENDLOOP
 ENDLOOP
 RETURN records
END
```

48. get_column_index() function:

```
DEF get_column_index(col_name, records):
BEGIN
 DECARE header
 header = records[0]
 LOOP i FROM 0 TO len(header) STEP BY 1:
```

```
  IF (header[i] == col_name):
   RETURN i
  ENDIF
 ENDLOOP
END
```

49. select_column_from_table() function:

```
DEF select_columns_from_table(col_list, records):
BEGIN
 DECLARE indices, new_record, result, index
 indices = []
 LOOP i FROM 0 TO len(col_list) STEP BY 1:
  index = CALL get_column_index(col_list[i], records)
  CALL append index TO indices
 ENDLOOP
 result = []
 LOOP i FROM 0 TO len(records) STEP BY 1:
  new_record = []
  LOOP j FROM 0 TO len(indices) STEP BY 1:
   CALL append records[i][j] TO new_record
  ENDLOOP
  CALL append new_record TO result
 ENDLOOP
 RETURN result
END
```

50. select_columns_from_table() function:

```
DEF select_columns_from_table(col_list, records):
BEGIN
 DECLARE indices, new_record, result, index
 indices = []
 LOOP i FROM 0 TO len(col_list) STEP BY 1:
  index = CALL get_column_index(col_list[i], records)
  CALL append index TO indices
 ENDLOOP
 result = []
 LOOP i FROM 0 TO len(records) STEP BY 1:
  new_record = []
  LOOP j FROM 0 TO len(indices) STEP BY 1:
   CALL append records[i][j] TO new_record
  ENDLOOP
  CALL append new_record TO result
 ENDLOOP
 RETURN result
END
```

51. select_column_from_table_where() function:

```
DEF select_column_from_table_where(target_col, query_col, query, records):
BEGIN
 DECLARE target_index, query_index, result
 target_index = CALL get_column_index(target_col, records)
 query_index = CALL get_column_index(query_col, records)
 result = []
 LOOP i FROM 0 TO len(records) STEP BY 1:
  IF (records[i][query_index] == query):
   CALL append records[i][target_index] TO result
  ENDIF
 ENDLOOP
 RETURN result
END
```

52. select_row_from_table_where() function:

```
DEF select_row_from_table_where(col_name, query, records):
BEGIN
 DECLARE index, result
 index = CALL get_column_index(col_name, records)
 result = [records[0]]
 LOOP i FROM 0 TO len(records) STEP BY 1:
  IF (records[i][index] == query):
   CALL append records[i] TO result
  ENDIF
 ENDLOOP
 RETURN result
END
```

53. write_record() function:

```
DEF write_record(record, filename, interactive=True):
BEGIN
 CALL split on record
 OPEN filename MODE "a"
  WRITE record TO file
 CLOSE file
 IF (interactive == True):
  DISPLAY "Succesfully added record. Press enter to continue: "
  READ
 ENDIF
END
```

54. write_records() function:

```
DEF write_records(records, filename, interactive=True):
BEGIN
 DECLARE towrite
 LOOP i FROM 0 TO len(records) STEP BY 1:
  LOOP j FROM 0 TO len(records[i]) STEP BY 1:
   IF (records[i][j] == "-"):
    records[i][j] == ""
   ENDIF
  ENDLOOP
  towrite = ""
  CALL join records[i] TO towrite
  records[i] = towrite
 ENDLOOP
 OPEN filename MODE "w"
  WRITE lines of records TO file
 CLOSE file
 IF (interactive == True):
  DISPLAY "Succesfully modified record. Press enter to continue: "
  READ
 ENDIF
END
```

55. bubble_sort_records() function:

```
DEF bubble_sort_records(col_name, records):
BEGIN
 DECLARE index, sorted_records, n, temp
 index = get_column_index(col_name, records)
 sorted_records = [records[0]]
 records = records[1:]
 n = CALL len(records)
```

```
 LOOP i FROM 0 TO (n - 1) STEP BY 1:
  LOOP j FROM 0 TO (n - i - 1) STEP BY 1:
   IF (Records[j][index] > records[j + 1][index]):
    temp = records[j + 1]
    records[j + 1] = records[j]
    records[j] = temp
   ENDIF
  ENDLOOP
 ENDLOOP
 sorted_records = sorted_records + records
 RETURN sorted_records
END
```

56. reverse_records() function:

```
DEF reverse_records(records):
BEGIN
 DECLARE result
 result = [records[0]]
 LOOP i from (len(records) - 1) TO 1 STEB BY -1:
  CALL append records[i] TO result
 ENDLOOP
 RETURN result
END
```

57. modify_record() function:

```
DEF modify_record(new_record, records):
BEGIN
 DECLARE modified_records
 modified_records = records
 LOOP i FROM 1 TO len(records) STEP BY 1:
  IF (records[i][0] == new_record[0]):
   LOOP j FROM 1 TO len(new_record):
    IF (new_record[j] != ""):
     modified_records[i][j] = new_record[j]
    ENDIF
   BREAK
  ENDIF
 ENDLOOP
 RETURN modified_records
END
```

58. left_join_records_on function:

```
DEF left_join_records_on(records1, col1, records2, col2):
BEGIN
 DECLARE col1_index, col2_index, result
 col1_index = CALL get_column_index(col1, records1)
 col2_index = CALL get_column_index(col2, records2)
 result = [records1[0] + records2[0]]
 LOOP i FROM 1 TO len(records1):
  LOOP j FROM 1 TO len(records2):
   IF (records1[i][col1_index] == records2[j][col2_index]):
    CALL append (records1[i] + records[2])
   ENDIF
  ENDLOOP
 ENDLOOP
 RETURN result
END
```

59. hhmm_to_seconds() function:

```
DEF hhmm_to_seconds(time):
BEGIN
 DECLARE result
 result = time[:2] * 3600 + time[3:] * 60
 RETURN result
END
```

60. get_date() function:

```
DEF hash_password(password):
BEGIN
 DECLARE result
 result = CALL hashlib.md5(password)
 RETURN result
END
```

61. hash_password() function:

```
DEF hash_password(password):
BEGIN
 DECLARE result
 result = CALL hashlib.md5(password)
 RETURN result
END
```

62. int_list_sum() function:

```
DEF int_list_sum(ls):
BEGIN
 DECLARE result
 result = 0
 LOOP i FROM 0 TO len(ls) STEP BY 1:
  result += i
 ENDLOOP
 RETURN result
END
```

63. get_pretty_schedule() function:

```
DEF get_pretty_schedule(booked=False):
BEGIN
 DECLARE schedule_records, coach_records, sport_records, sport_centre_records
 DECLARE compound_records, columns
 schedule_records = CALL read_csv('data/schedule.txt')
 coach_records = CALL read_csv('data/coach.txt')
 sport_records = CALL read_csv('data/sport.txt')
 sport_centre_records = CALL read_csv('data/sport-centre.txt')
 compound_records = CALL left_join_records_on(schedule_records, 'schedule_coach_id', coach_records, 'coach_id')
 compound_records = CALL left_join_records_on(compound_records, 'coach_sport_id', sport_records, 'sport_id')
 compound_records = CALL left_join_records_on(compound_records, 'coach_sport_centre_id', sport_centre_records,
'sport_centre_id')
 columns = ['schedule_id', 'schedule_date', 'schedule_time',
        'schedule_duration', 'coach_name', 'sport_name',
        'sport_centre_name']
 IF (booked == True):
  CALL append "schedule_booked_ids" TO columns
 ENDIF
 records = CALL select_columns_from_table(columns, compound_records)
 RETURN records
END
```

64. pretty_print() function:

```
DEF pretty_print(records, interactive=True):
BEGIN
 DECLARE cols, longest_lines, length, sum
 cols = CALL len(records[0])
 longest_lines = []
 LOOP i FROM 0 TO len(cols) STEP BY 1:
  CALL append 0 to longest_lines
 ENDLOOP
 LOOP i FROM 0 TO len(records) STEP BY 1:
  LOOP j FROM 0 TO cols:
   length = CALL len(records[i][j])
   IF (length > longest_lines[j]):
    longest_lines[j] = length
   ENDIF
  ENDLOOP
 ENDLOOP
 LOOP i FROM 0 TO cols STEP BY 1:
  DISPLAY "| ", records[0][i], " " * (longest_lines[i] - len(records[0][i])) + " "
 ENDLOOP
 DISPLAY "|"
 sum = CALL int_list_sum(longest_lines)
 LOOP i FROM 0 TO (3 * COLS + 1 + sum):
  DISPLAY "-"
 ENDLOOP
 LOOP i FROM 1 TO len(records) STEP BY 1:
  LOOP j FROM 0 TO cols STEP BY 1:
   DISPLAY "| ", records[0][i], " " * (longest_lines[i] - len(records[0][i])) + " "
  ENDLOOP
  DISPLAY "|"
 ENDLOOP
 IF (interactive == True):
  DISPLAY "press enter to continue: "
 ENDIF
END
```

65. is_booked() function:

```
DEF is_booked(schedule_id, records):
BEGIN
 DECLARE schedule_id_index, booked_index
 schedule_id_index = CALL get_column_index('schedule_id', records)
 booked_index = CALL get_column_index('booked', records)
 LOOP i FROM 0 TO Len(records) STEP BY 1:
  IF (records[i][schedule_id_index] == schedule_id AND
    records[i][booked_index] == "yes"):
   RETURN True
  ENDIF
 ENDLOOP
 RETURN False
END
```

66. change_schedule() function:

```
DEF change_schedule(activity, schedule_id, records, student_id):
BEGIN
 DECLARE booked_index
 LOOP i FROM 0 TO len(records) STEP BY 1:
  IF (records[i][0] == schedule_id):
   new_cell = records[i][booked_index]
   CALL split on new_cell
   IF (activity == 'book'):
    CALL append student_id TO new_cell
   ELSE:
```

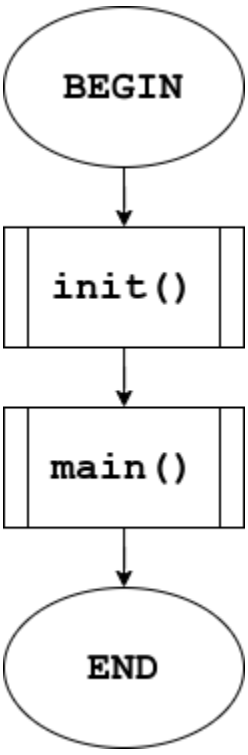
```
      CALL remove student_id TO new_cell
    ENDIF
    CALL join new_cell on records[i][booked_index]
  ENDIF
 ENDLOOP
 CALL write_records(records, "data/schedule.txt")
END
```

67. get_new_rating() function:

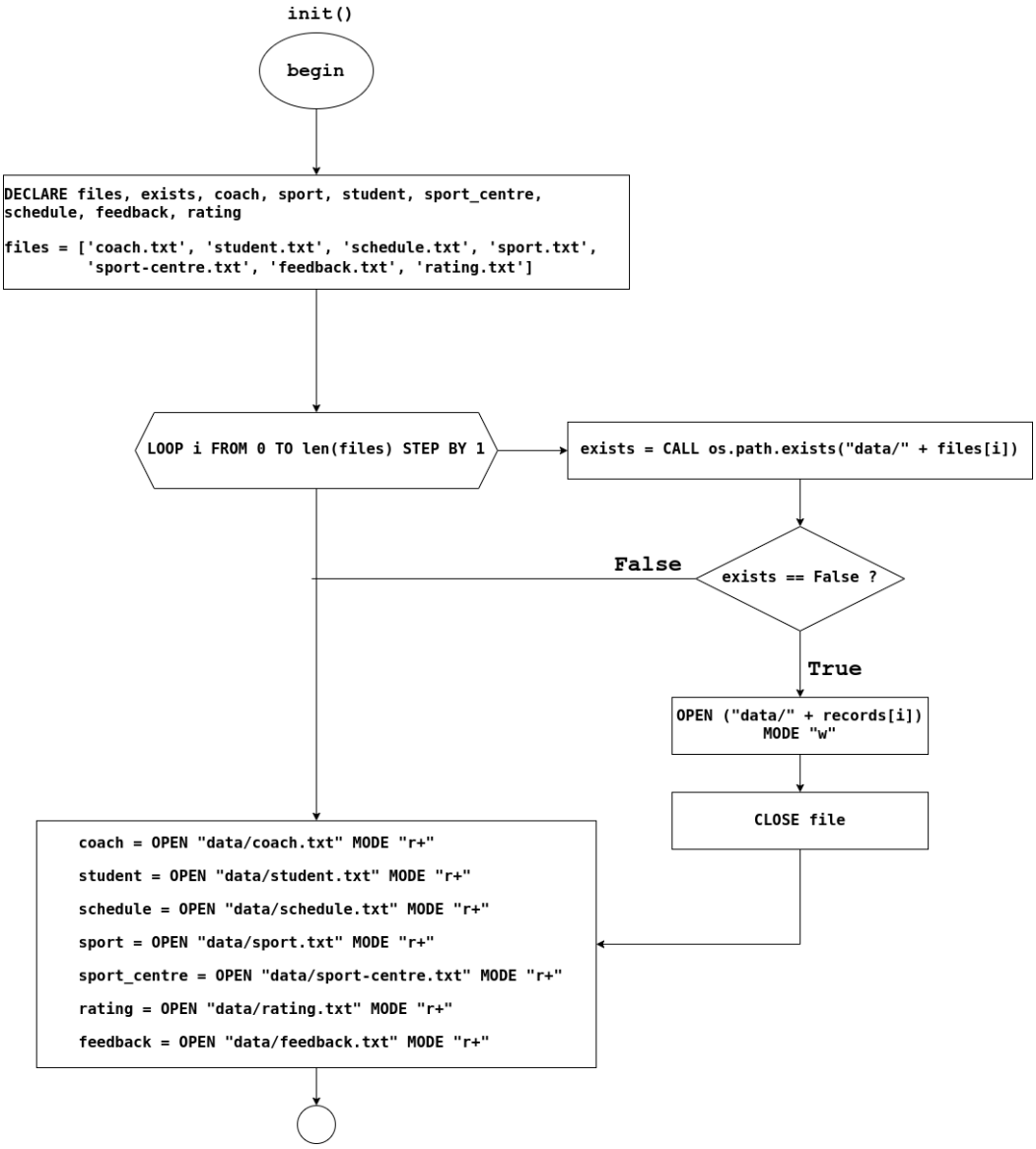
```
DEF get_new_rating(new_rating, old_rating, coach_rating, rating_count, rated):
BEGIN
 DECLARE rating
 IF (rated == False and coach_rating == 0):
  rating = new_rating
 ELIF (rated == False):
  rating = coach_rating + new_rating / 2
  CALL round(rating, 2)
 ELSE:
  rating = coach_rating * rating_count - old_rating
  rating = (rating + new_rating) / rating_count
  CALL round(rating, 2)
 ENDIF
 RETURN rating
END
```

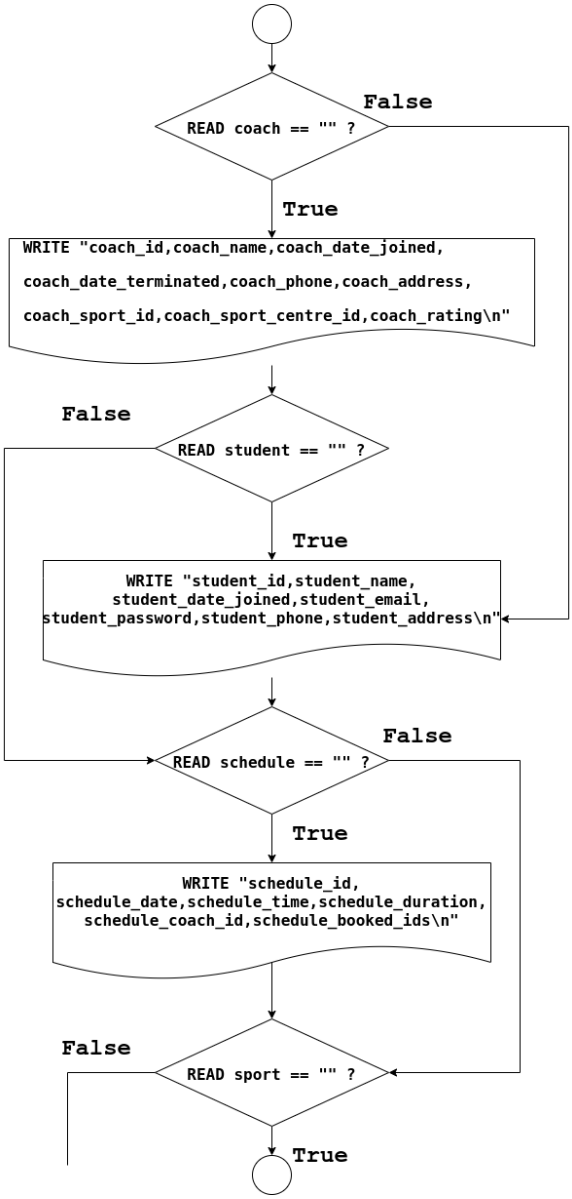
Design – Flowchart

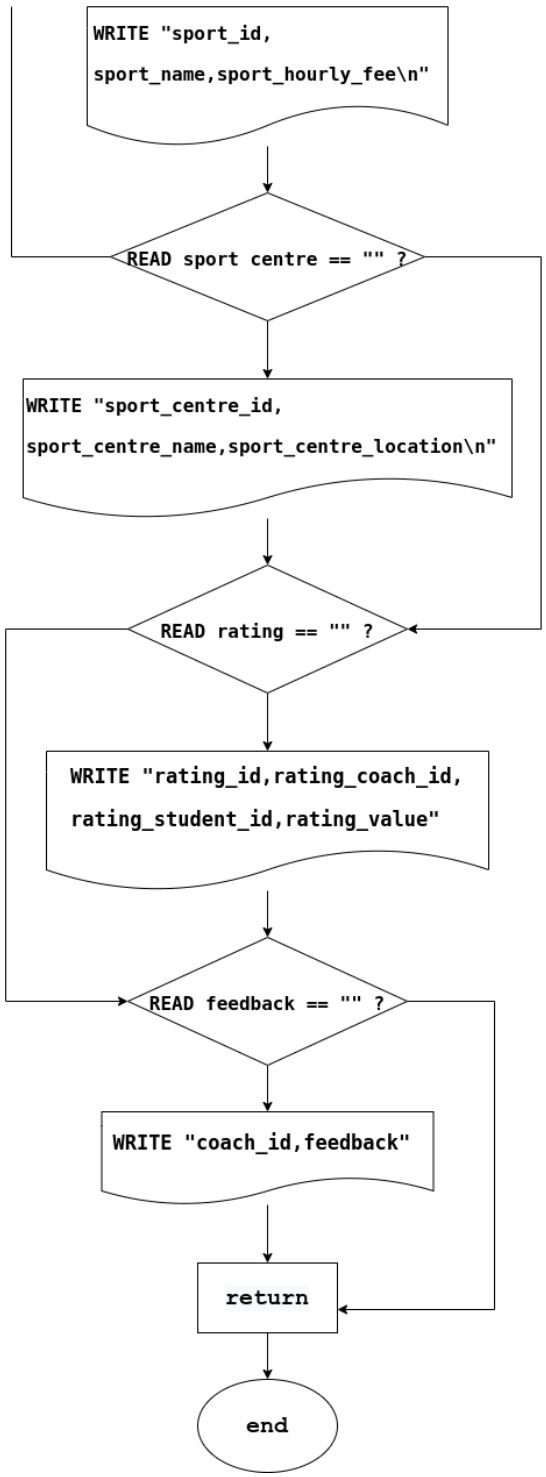
1. Program entry point



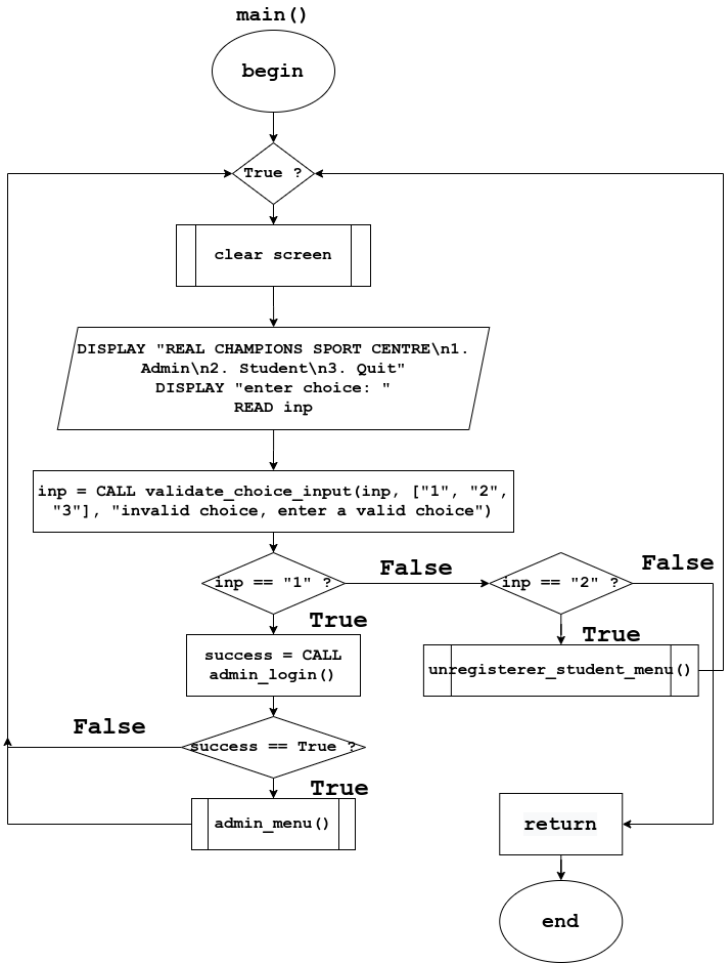
2. init() function



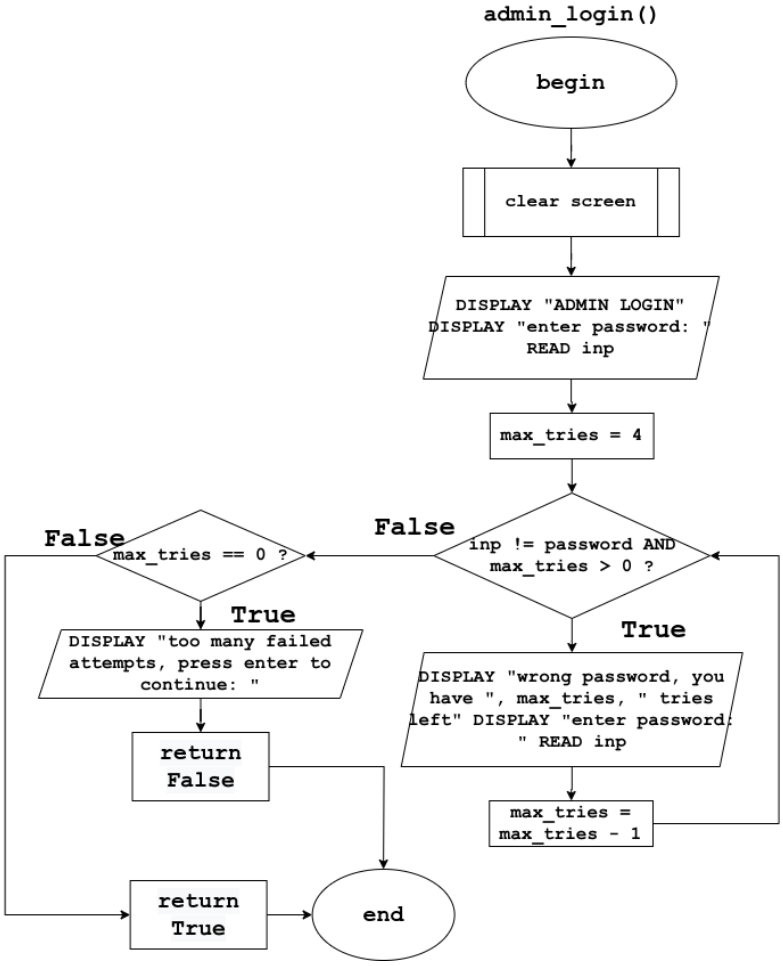

READ coach == "" ?
False
True
WRITE "coach_id,coach_name,coach_date_joined,
coach_date_terminated,coach_phone,coach_address,
coach_sport_id,coach_sport_centre_id,coach_rating\n"
READ student == "" ?
False
True
WRITE "student_id,student_name,
student_date_joined,student_email,
student_password,student_phone,student_address\n"
READ schedule == "" ?
False
True
WRITE "schedule_id,
schedule_date,schedule_time,schedule_duration,
schedule_coach_id,schedule_booked_ids\n"
READ sport == "" ?
False
True


WRITE "sport_id,
sport_name,sport_hourly_fee\n"
READ sport centre == "" ?
WRITE "sport_centre_id,
sport_centre_name,sport_centre_location\n"
READ rating == "" ?
WRITE "rating_id,rating_coach_id,
rating_student_id,rating_value"
READ feedback == "" ?
WRITE "coach_id,feedback"
return
end

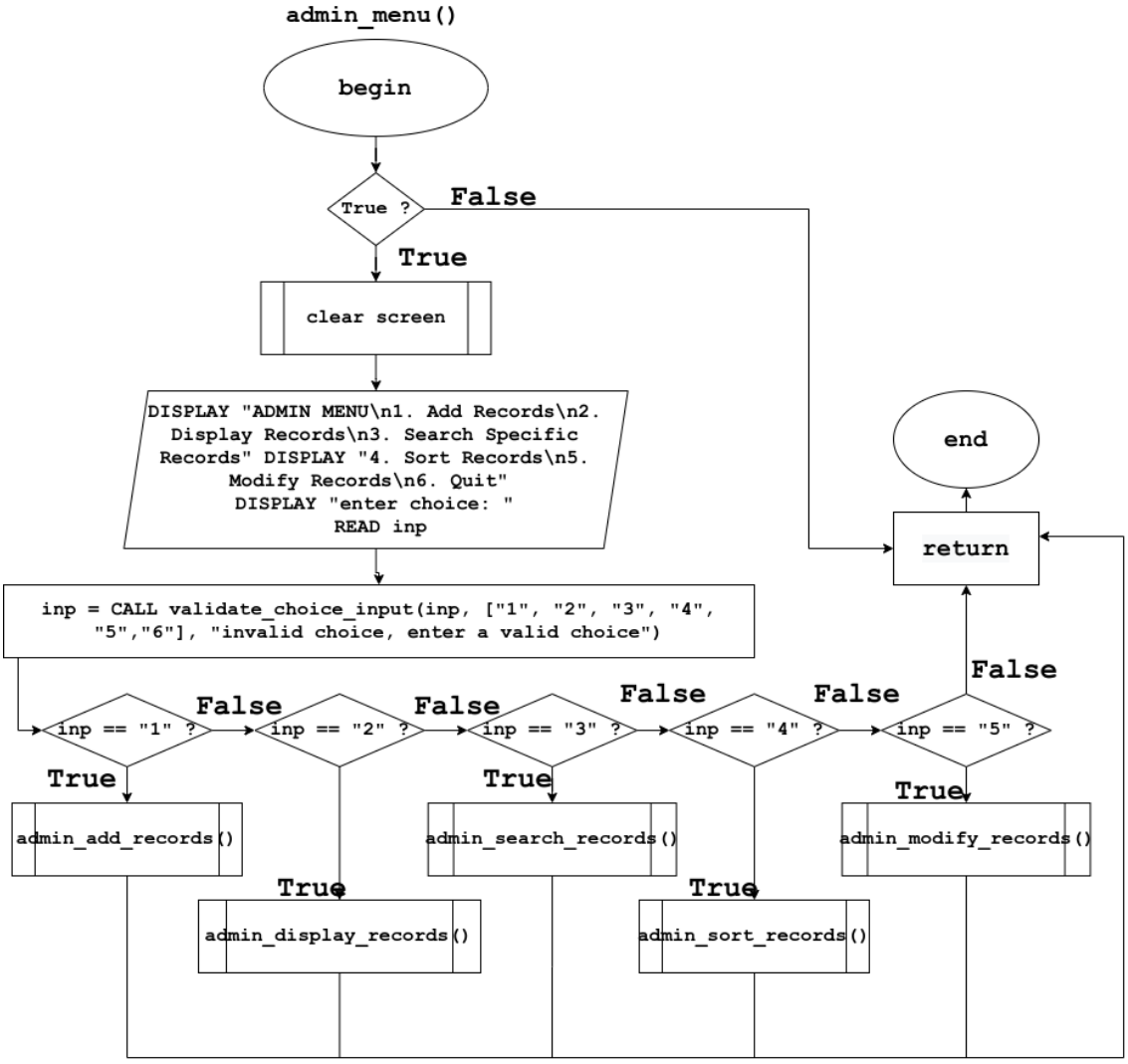
3. main() function



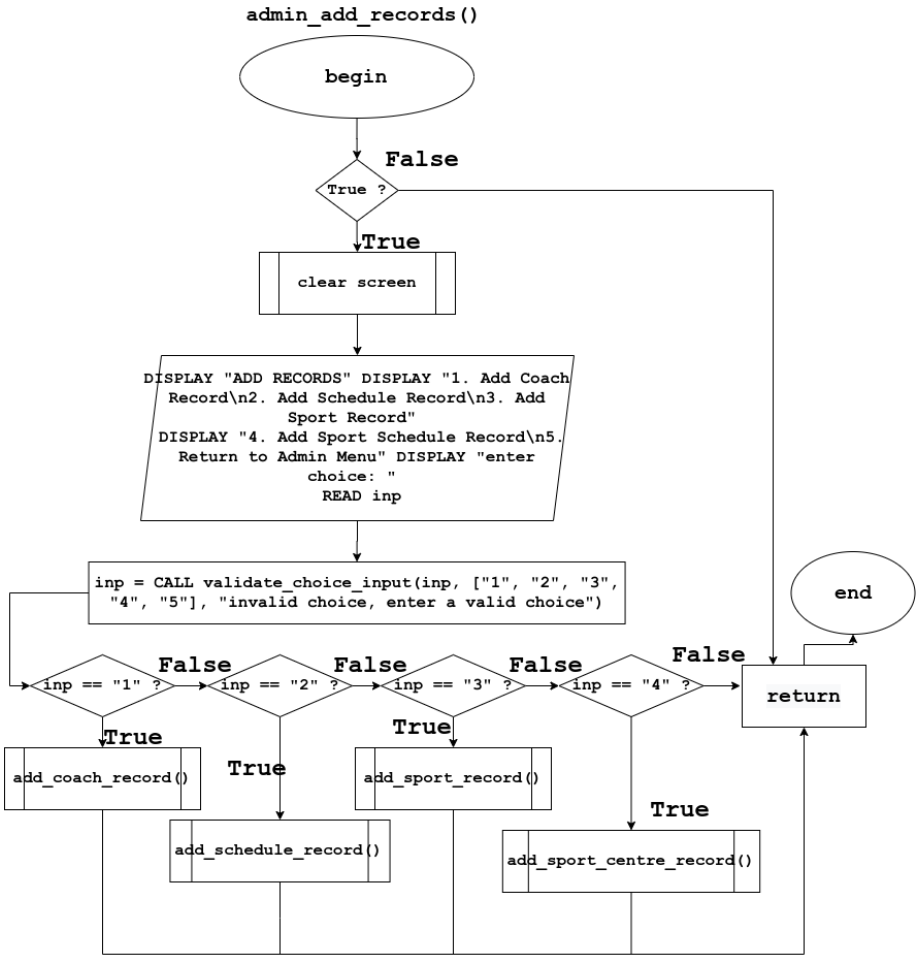
4. admin_login() function



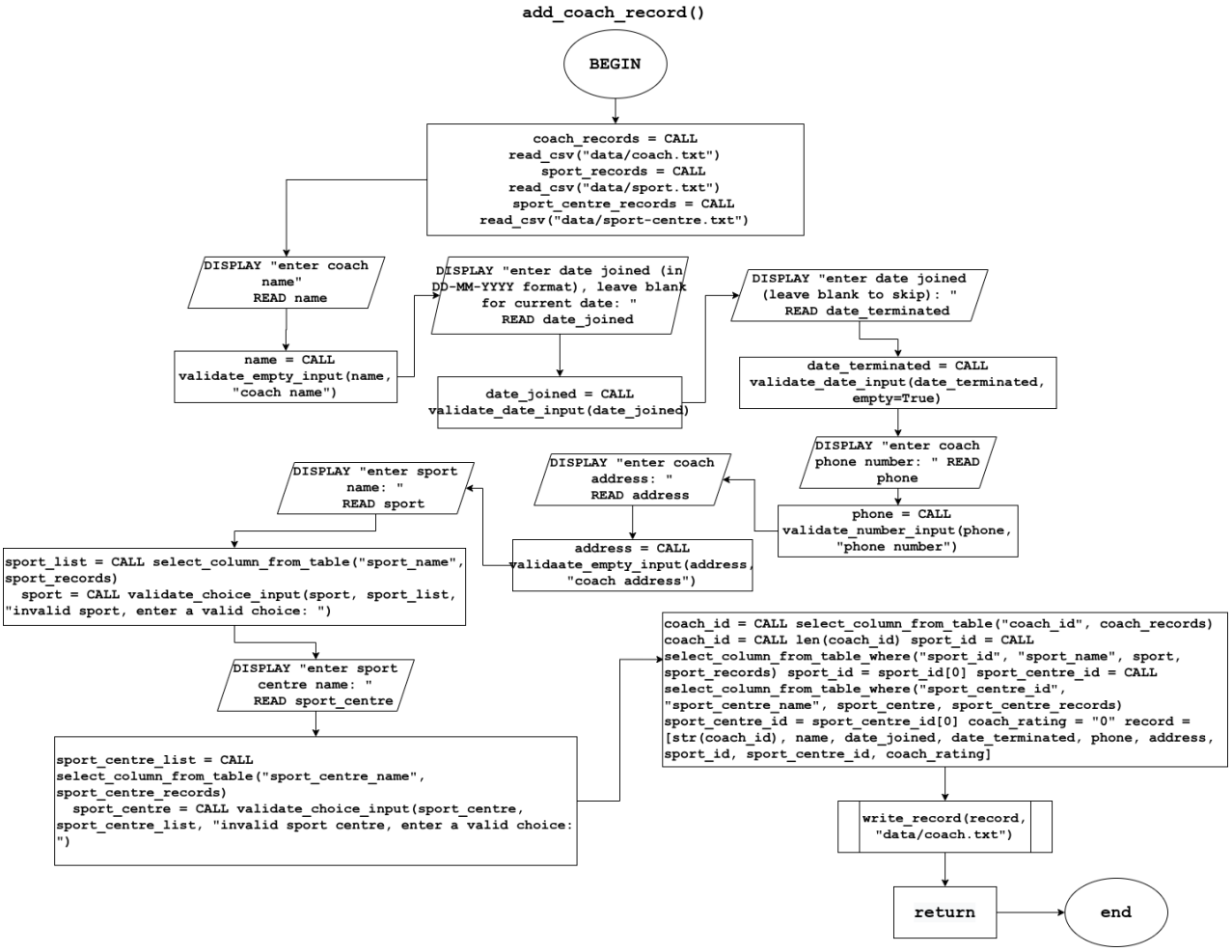
5. admin_menu() function



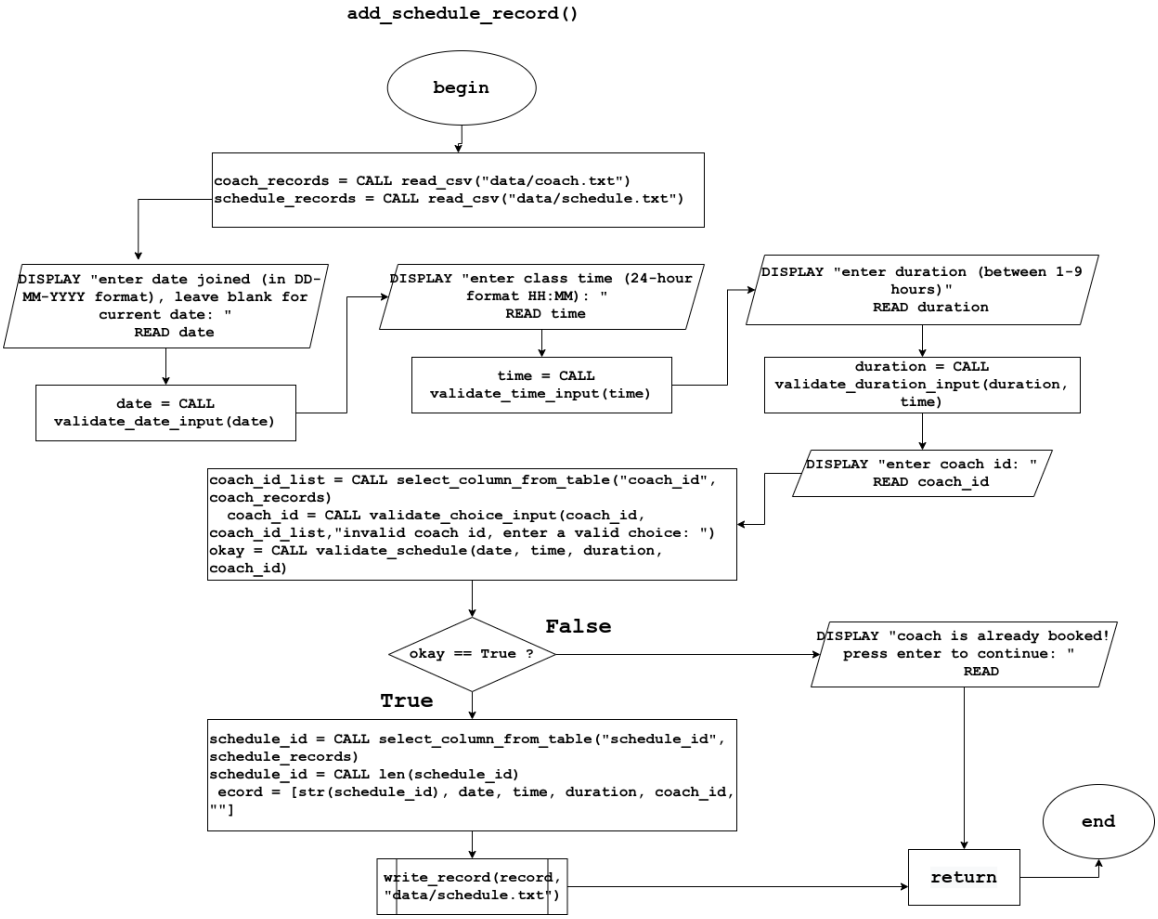
6. admin_add_records() function



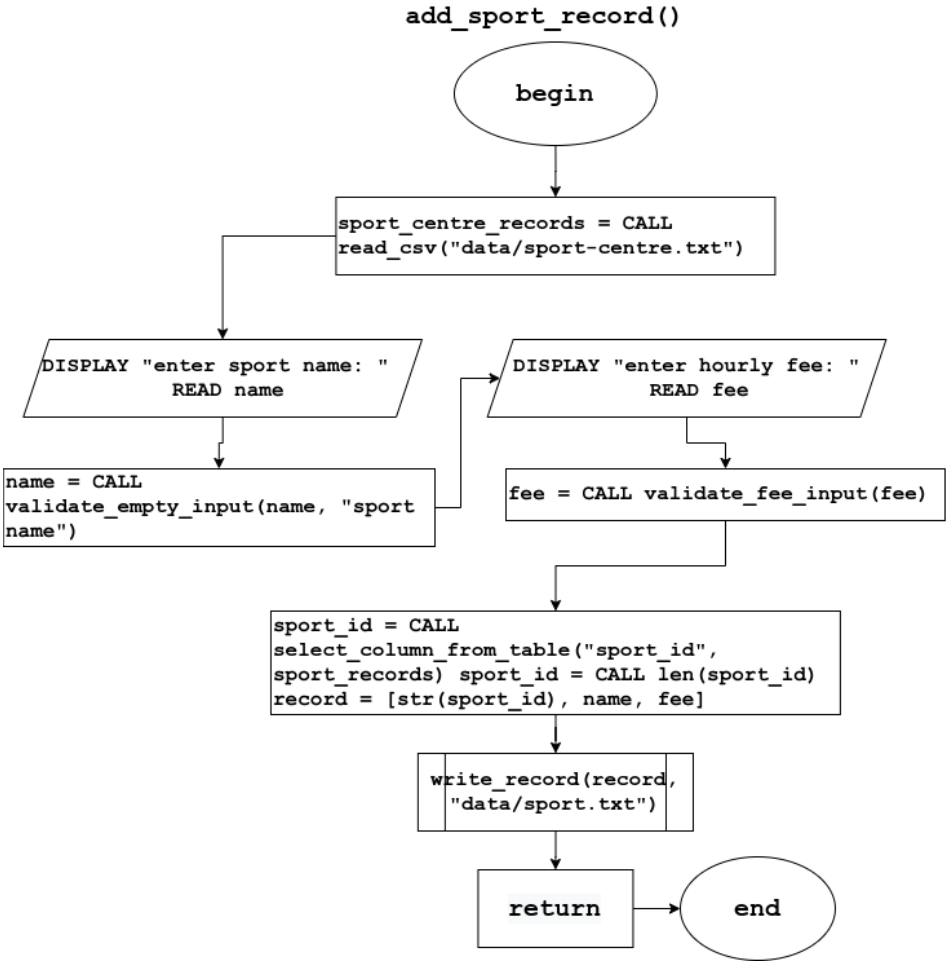
7. Add_coach_records() function



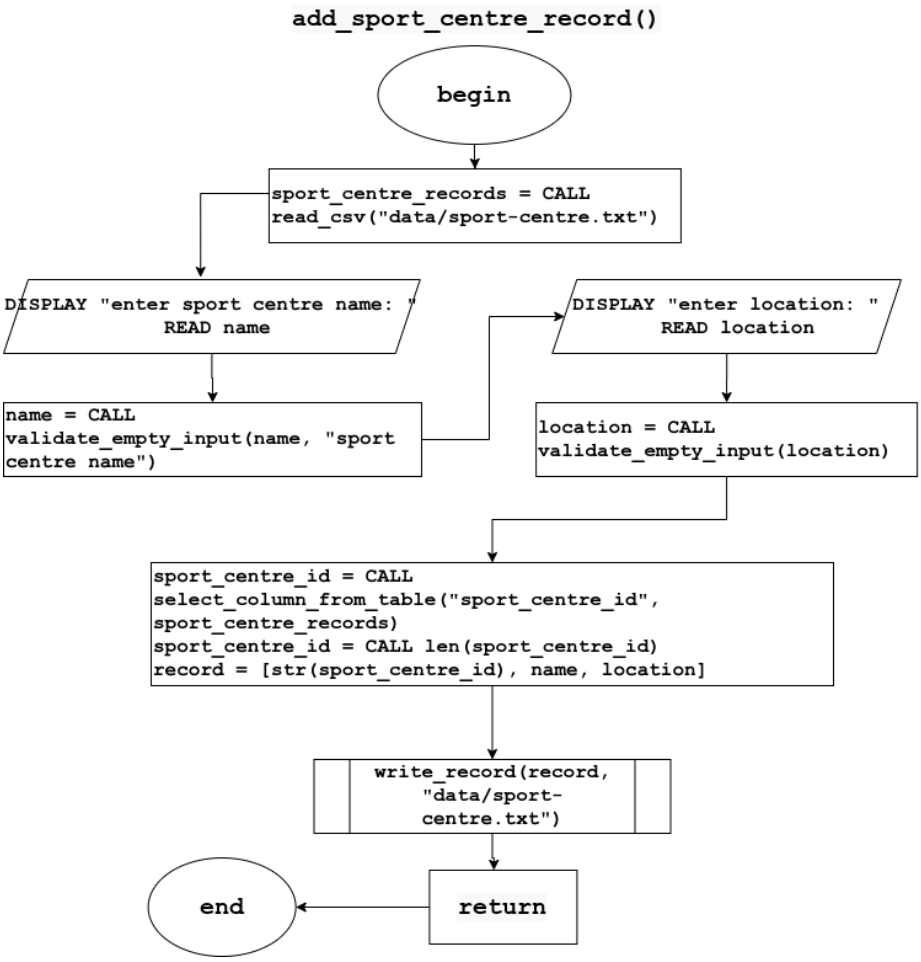
8. Add_schedule_records() function



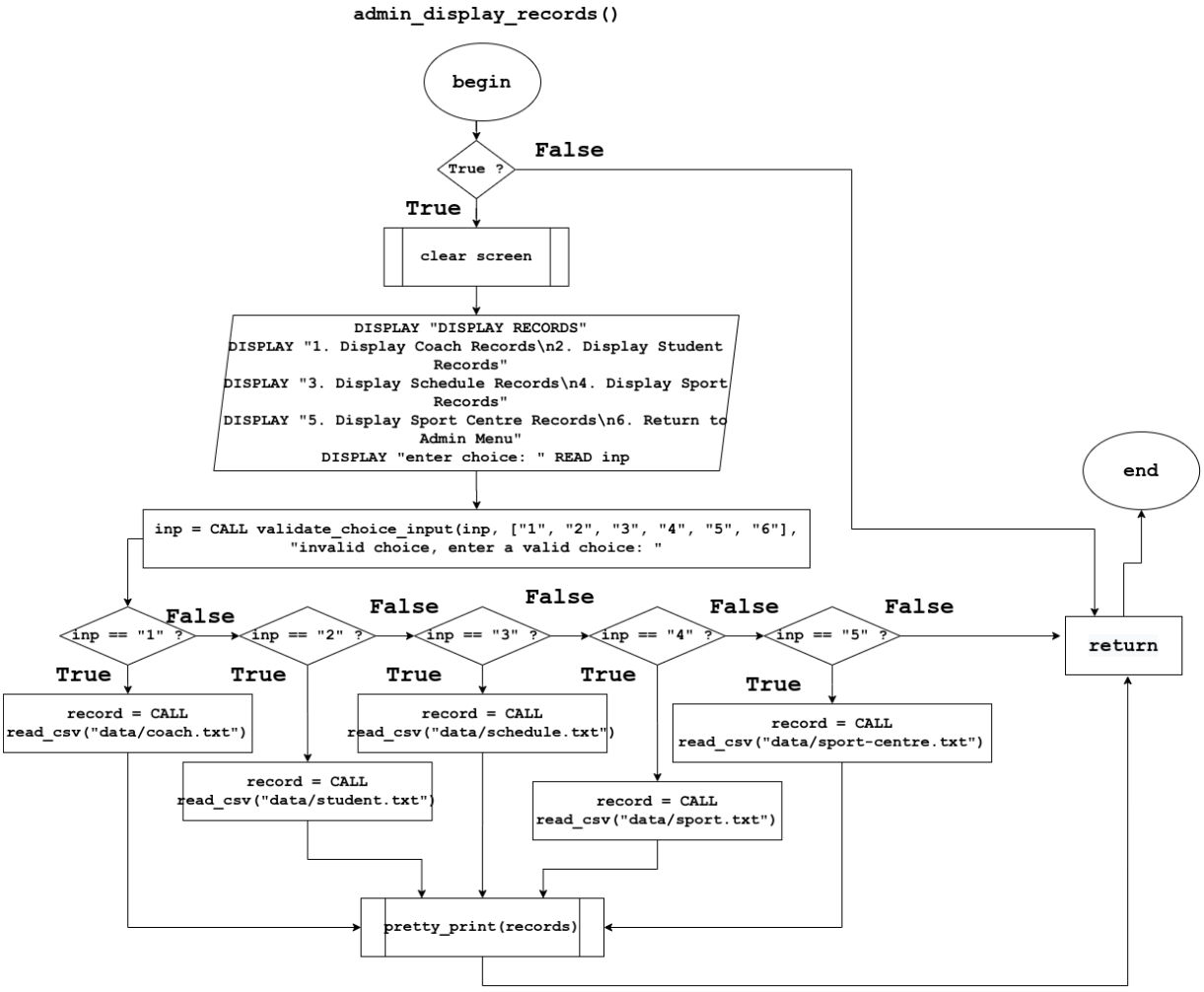
9. Add_sport_records() function



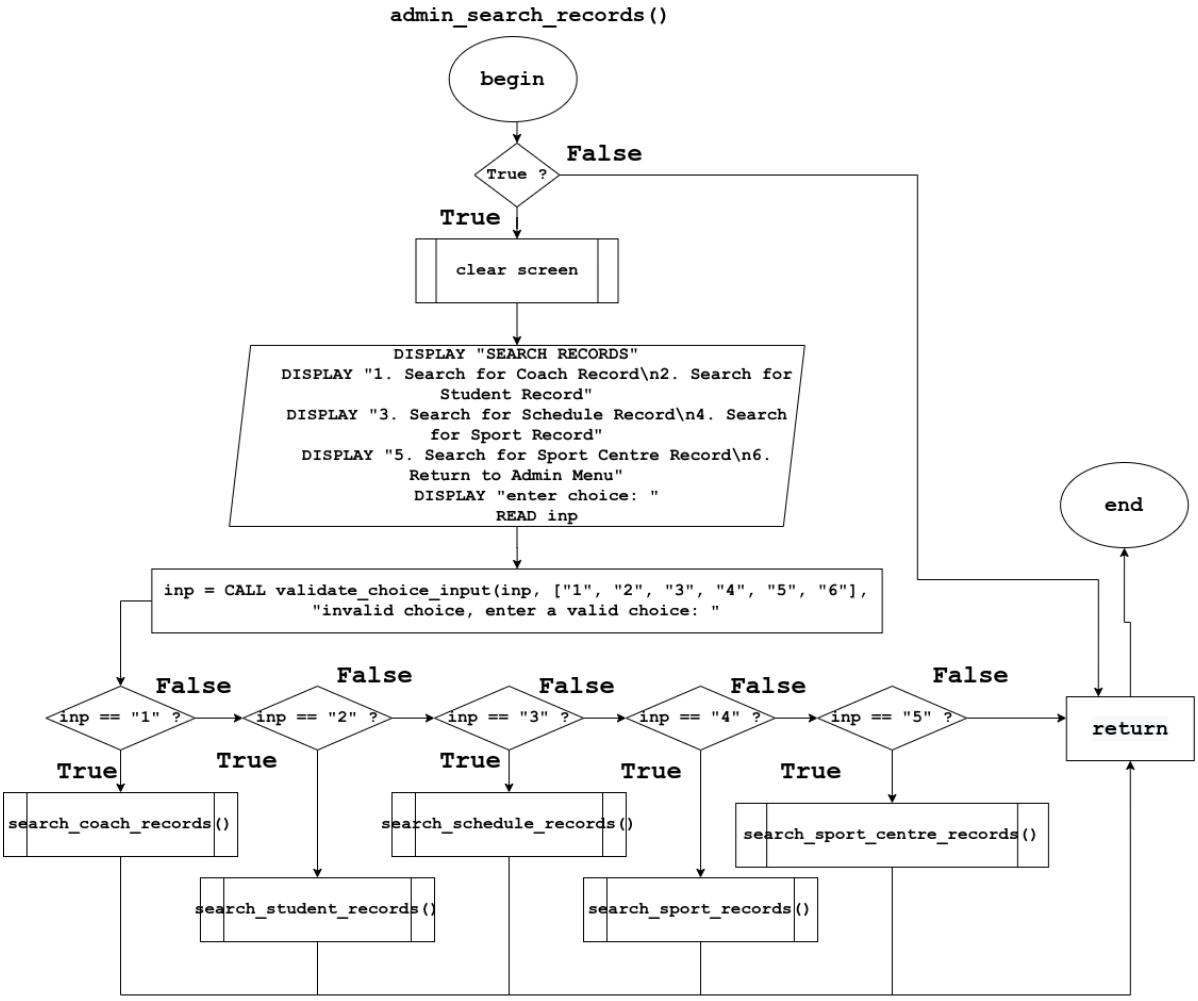
10. Add_sport_centre_records() function



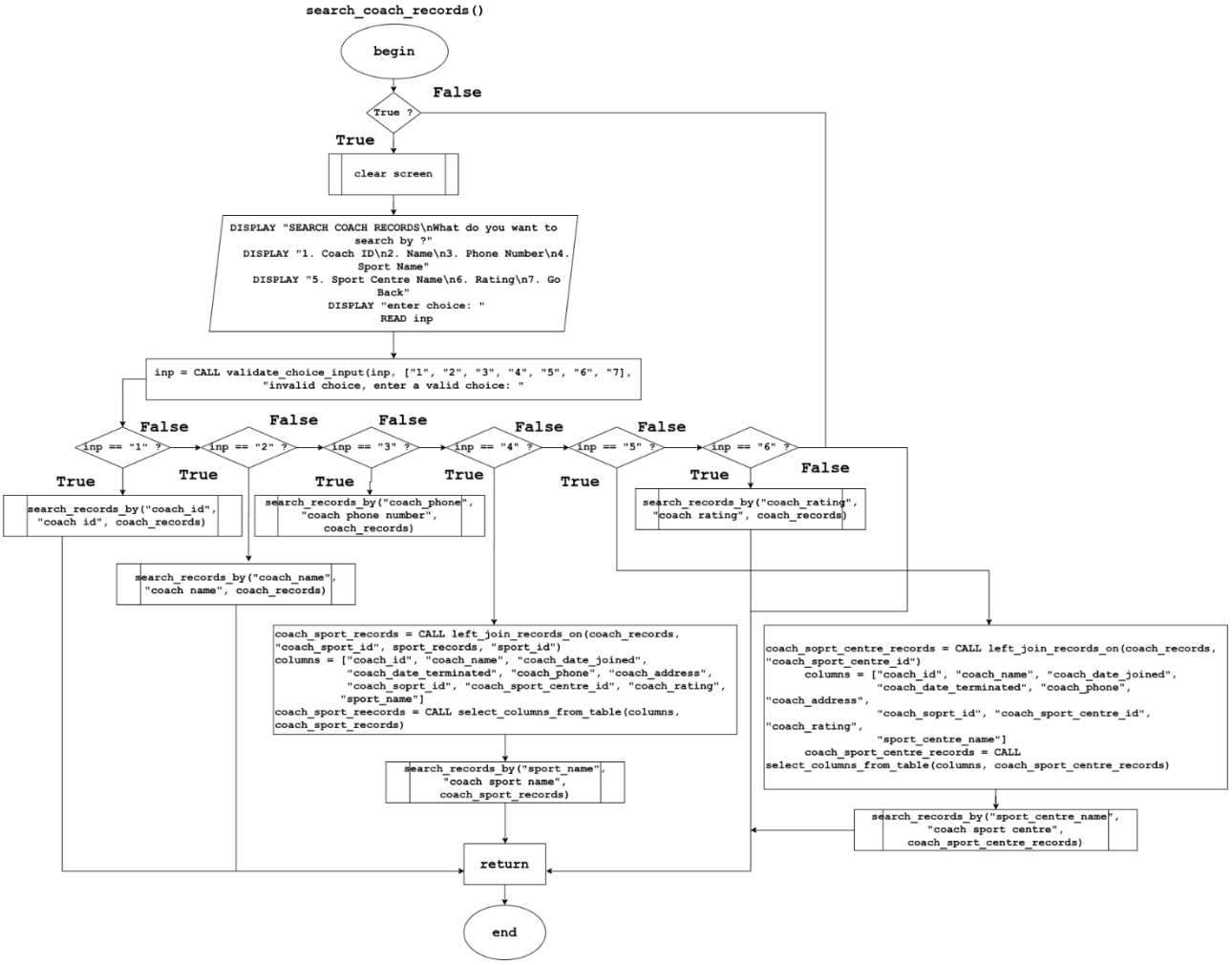
11. Admin_display_records()



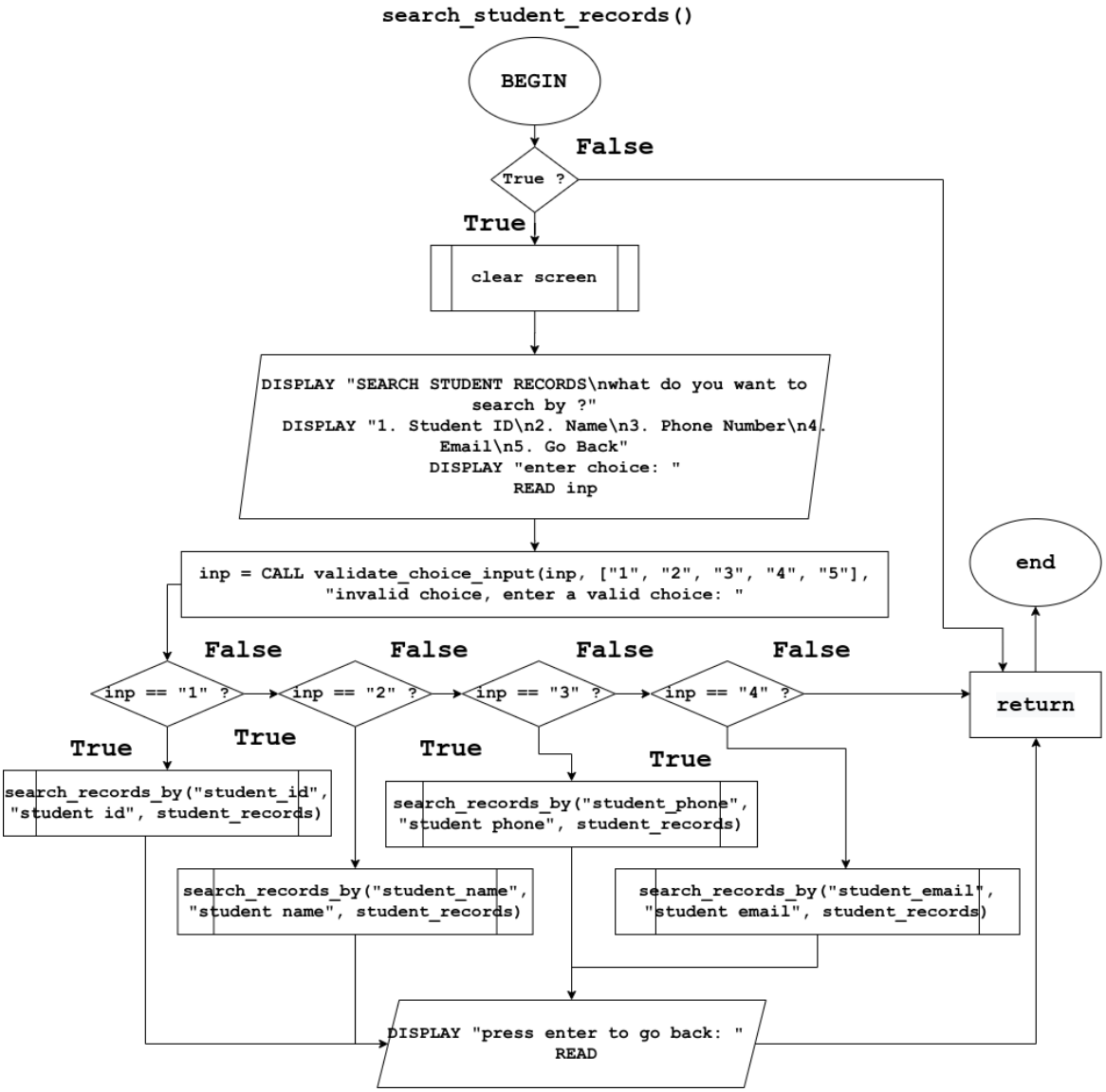
12. Admin_search_records()



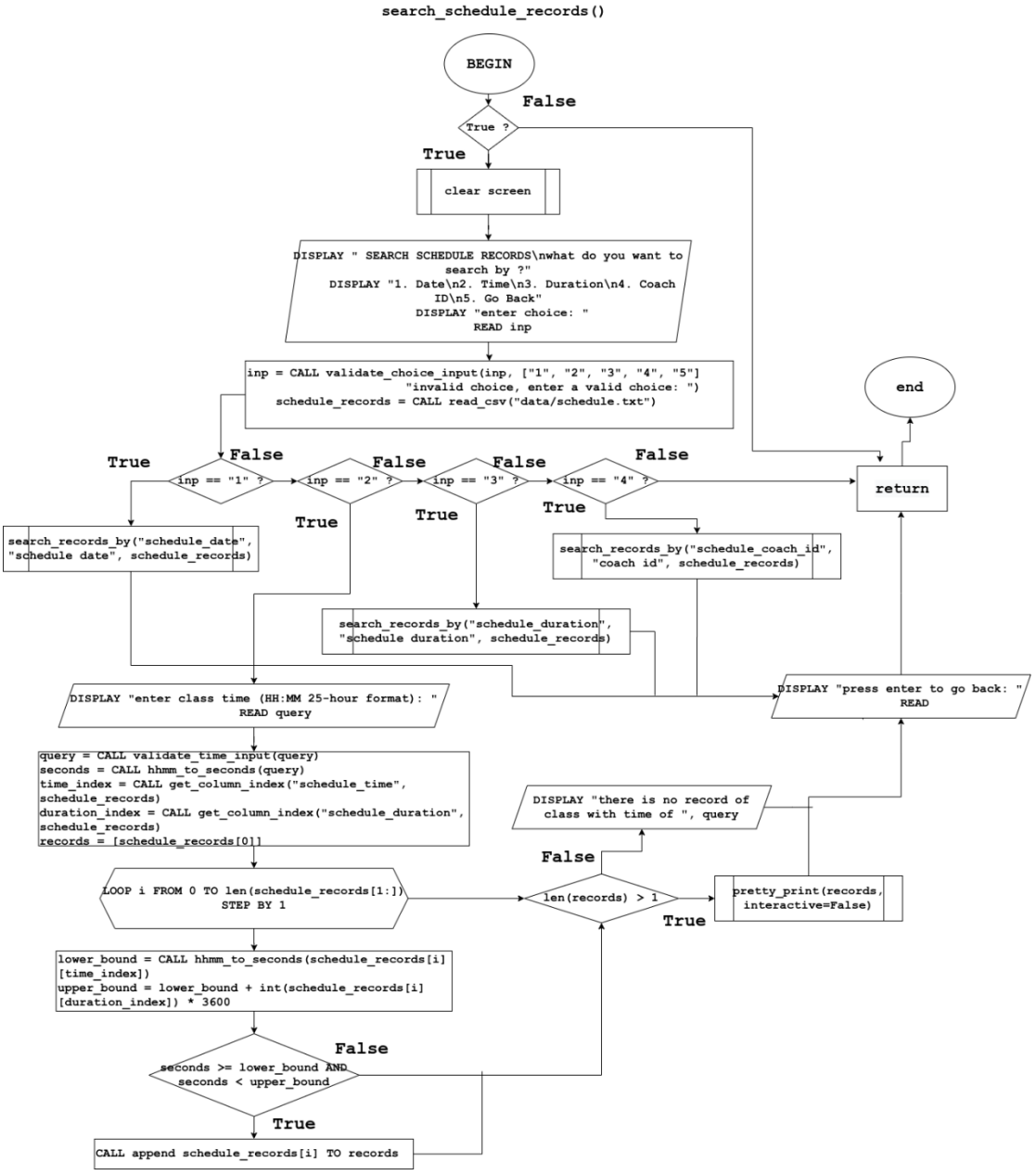
13. Search_coach_records()



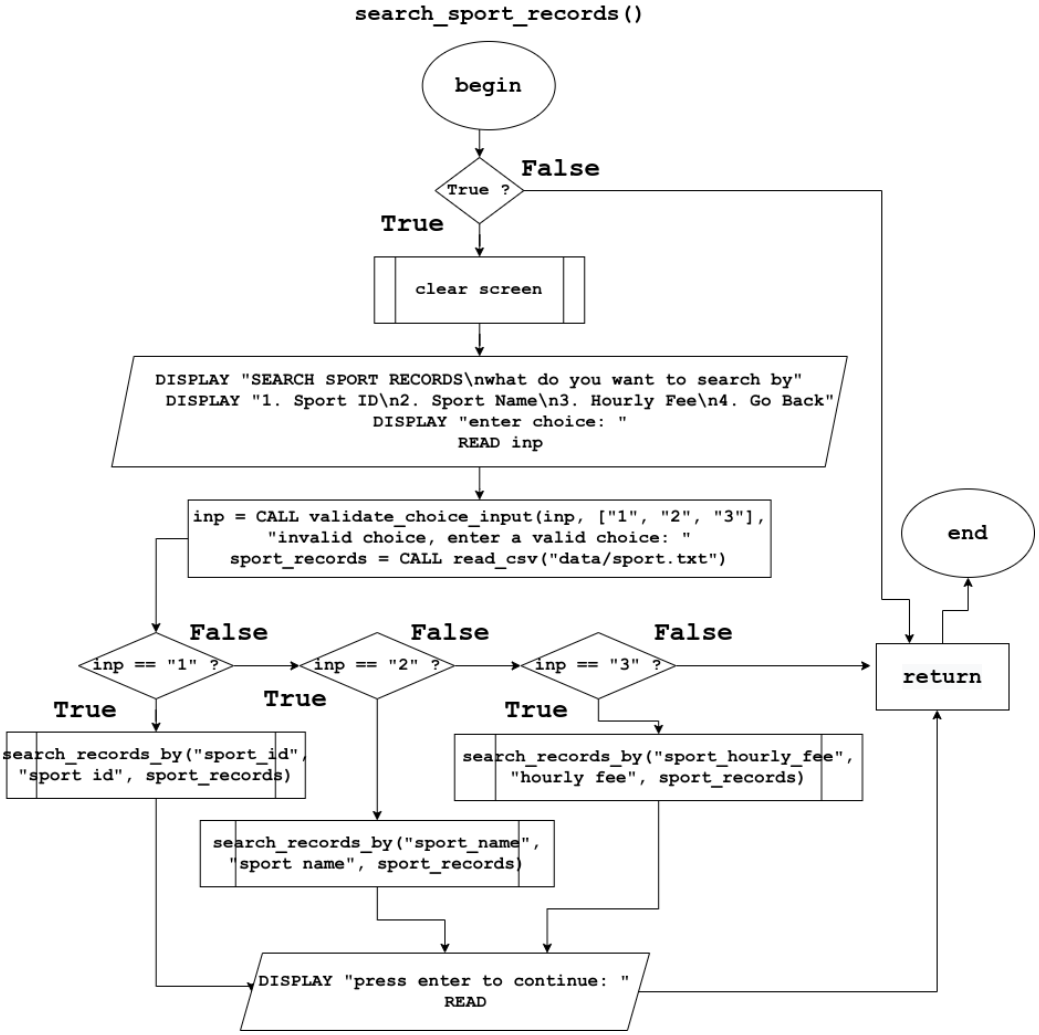
14. Search_student_records()



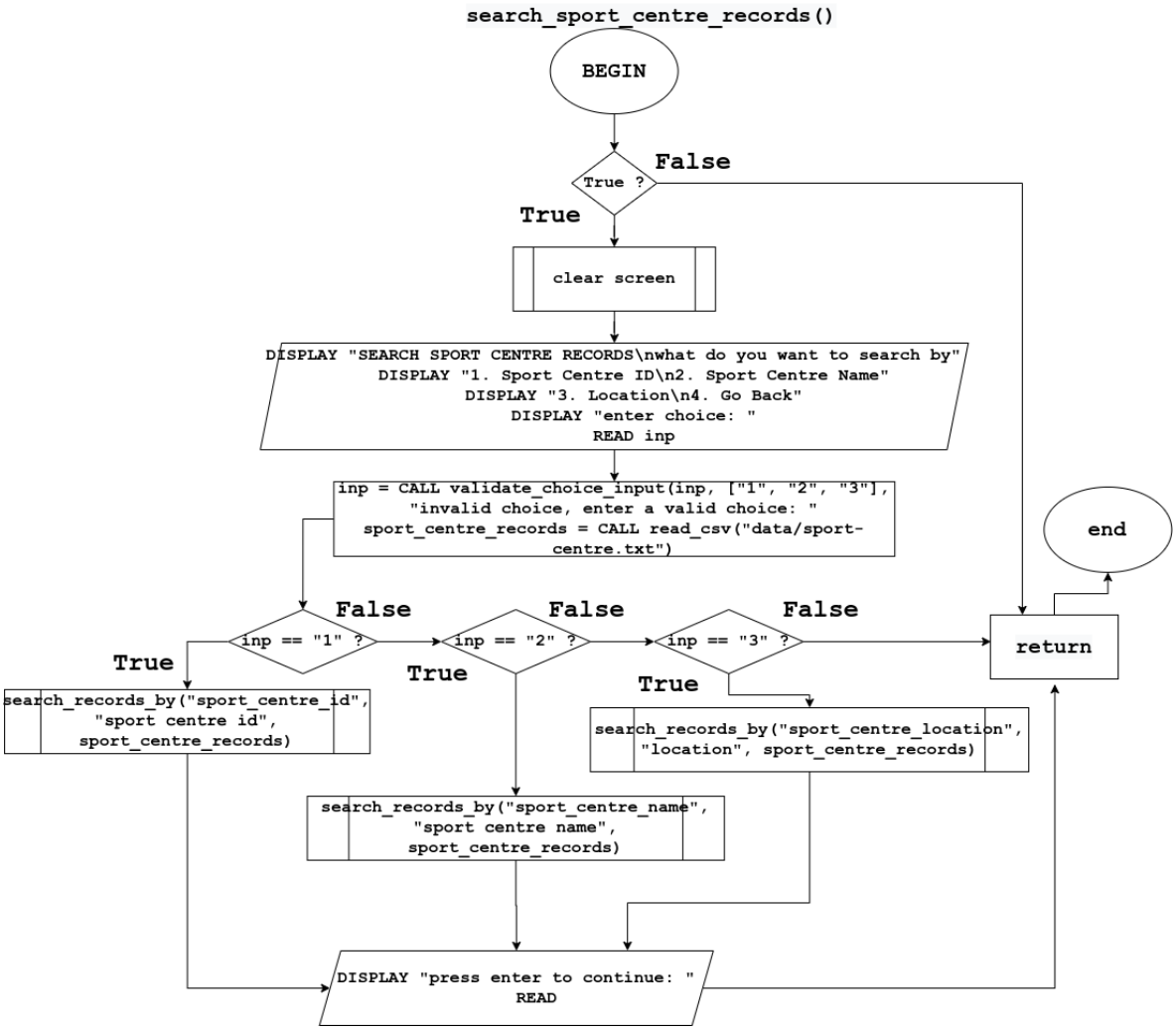
15. Search_schedule_records()



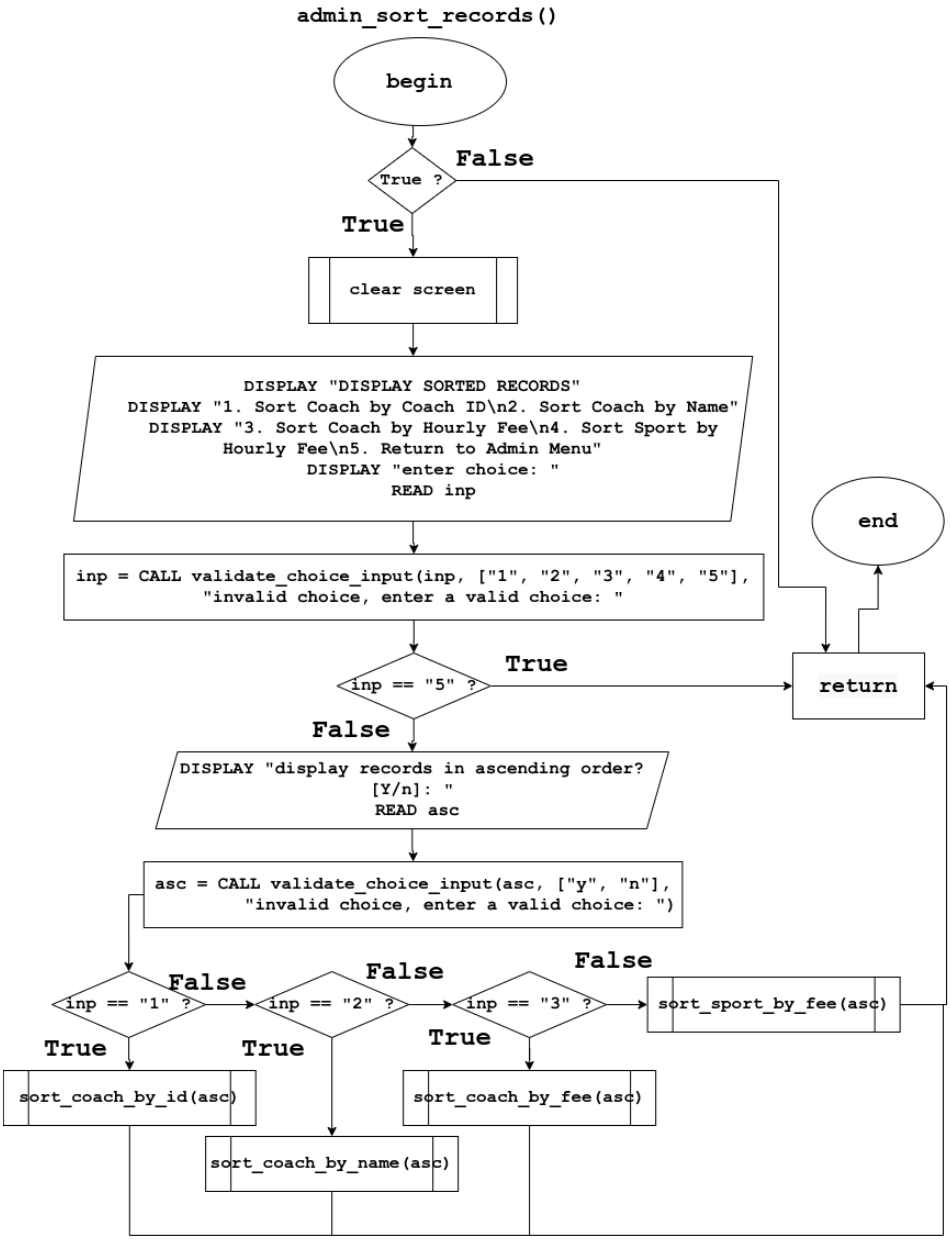
16. Search_sport_records()



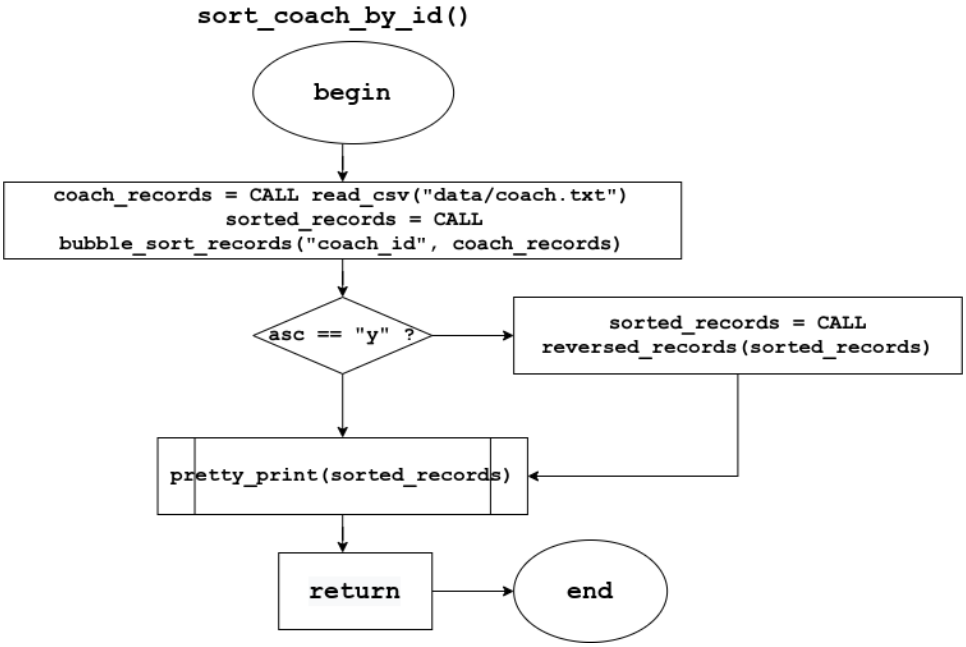
17. Search_sport_centre_records



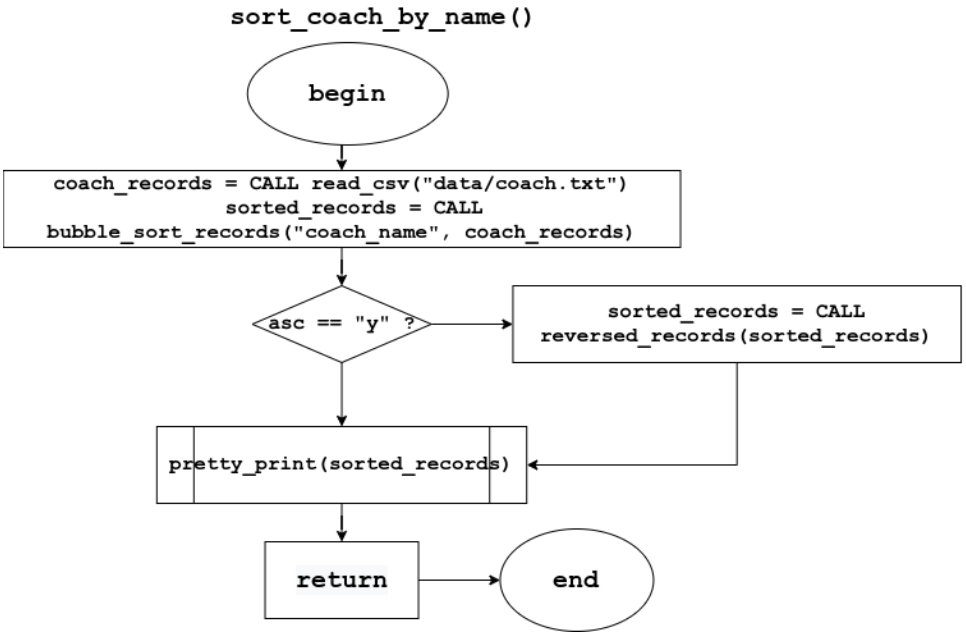
18. Admin_sort_records()



19. Sort_coach_by_id()

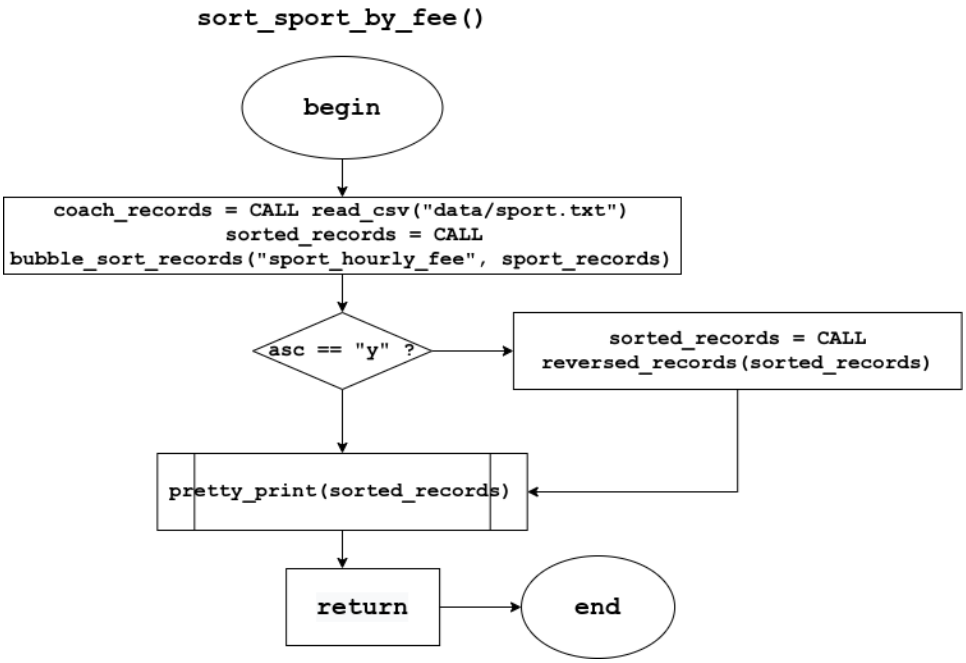


20. Sort_coach_by_name()

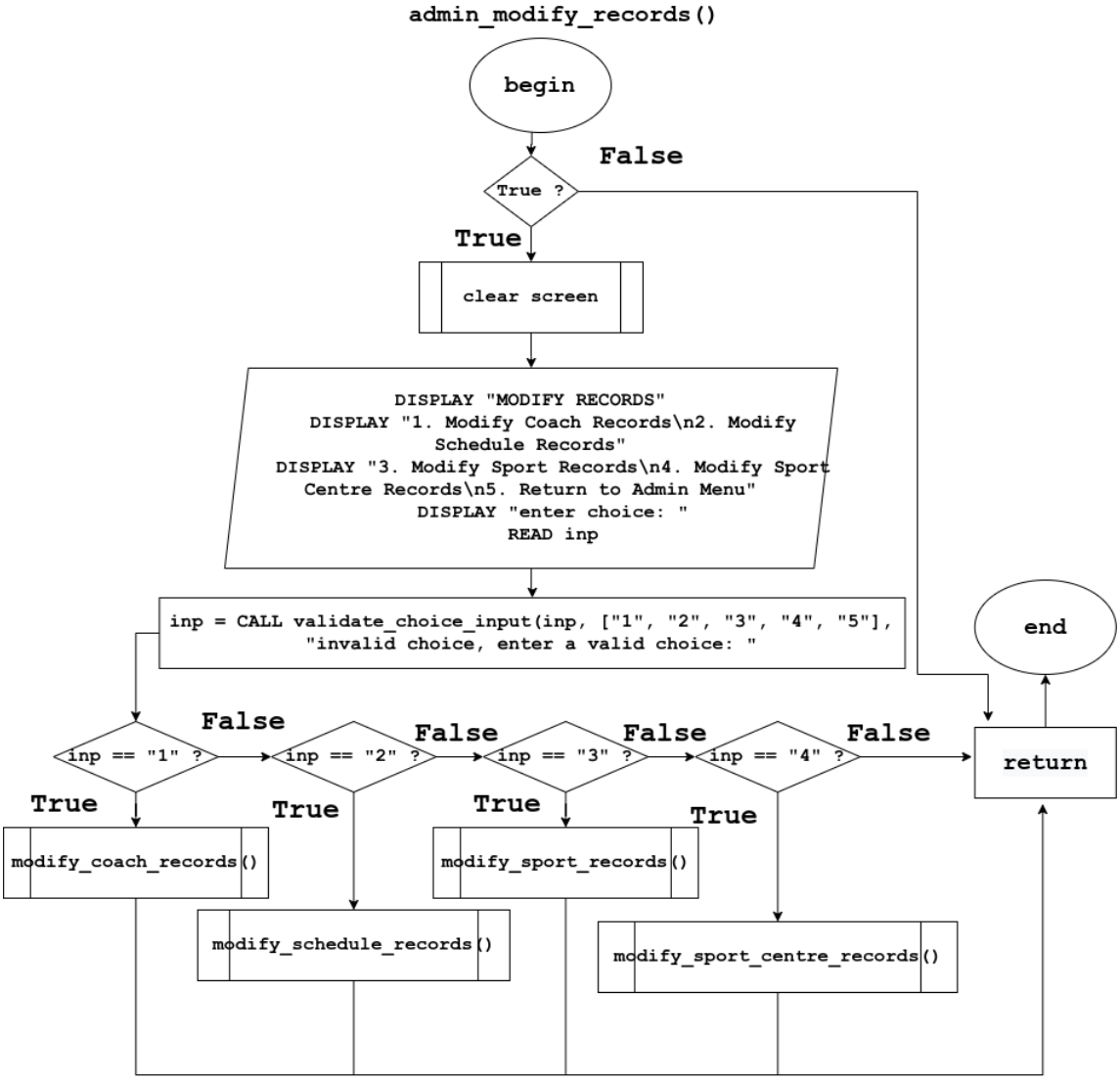


21. Sort_coach_by_fee()

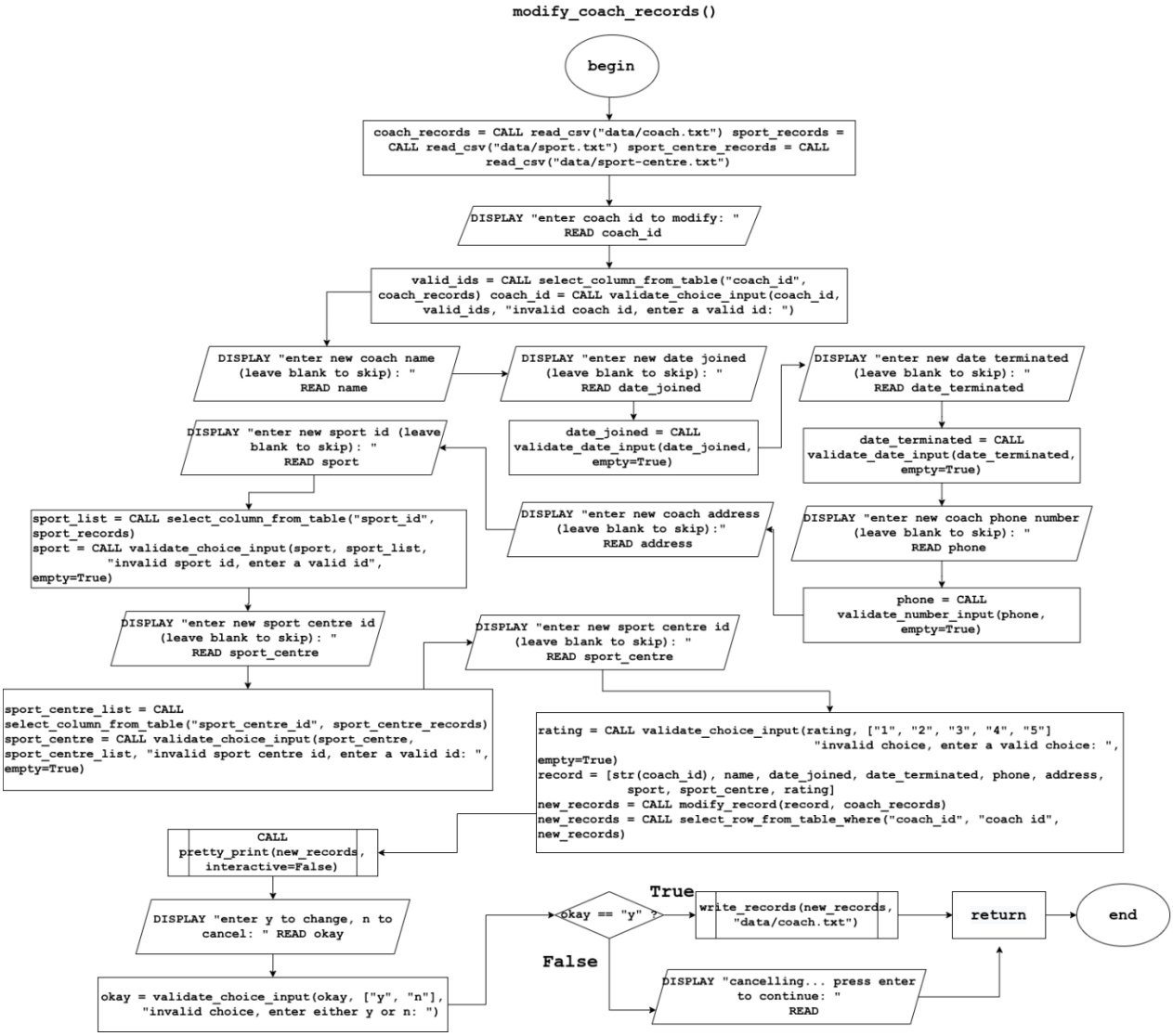
22. Sort_sport_by_fee()



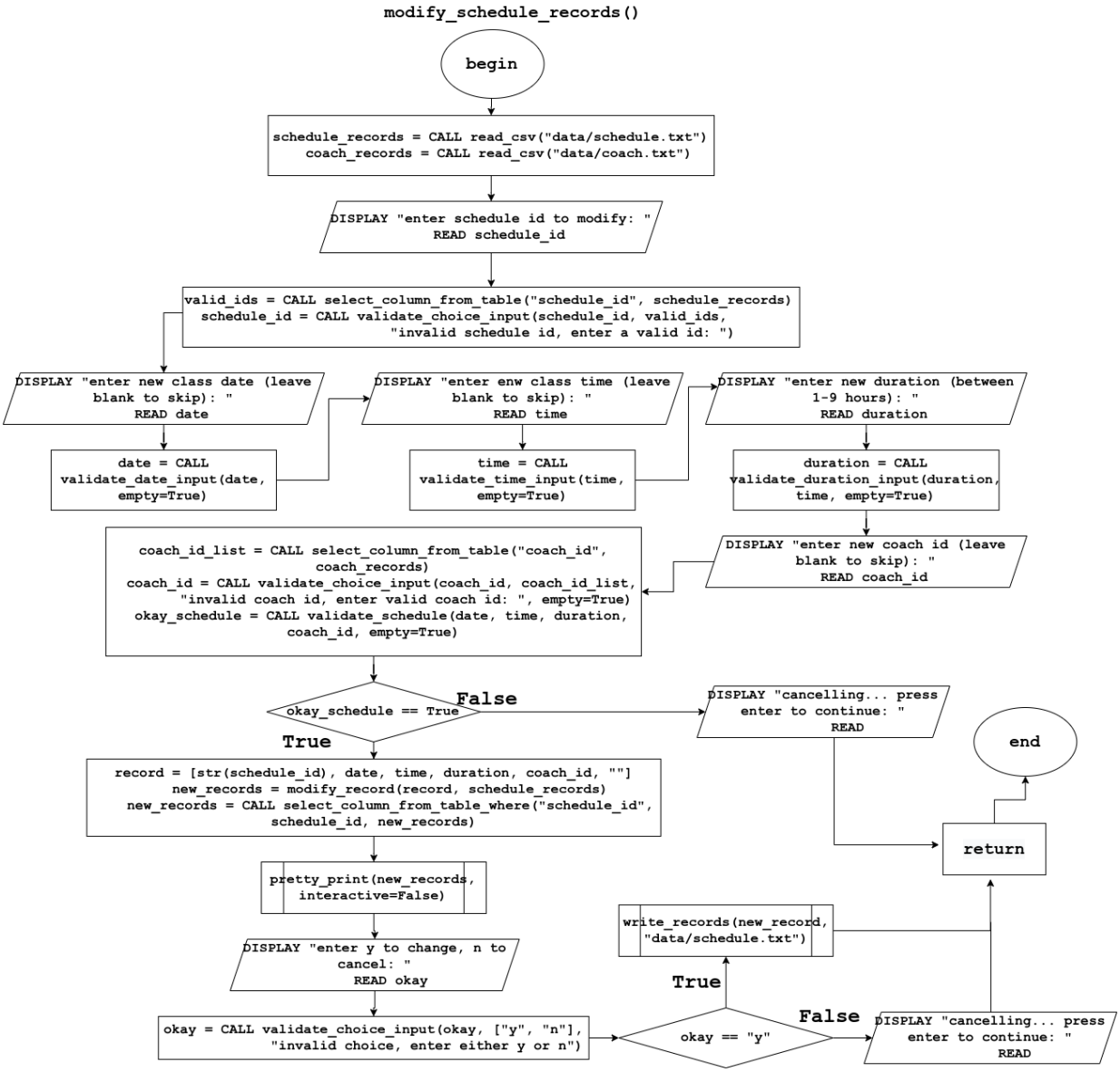
23. Admin_modify_records()



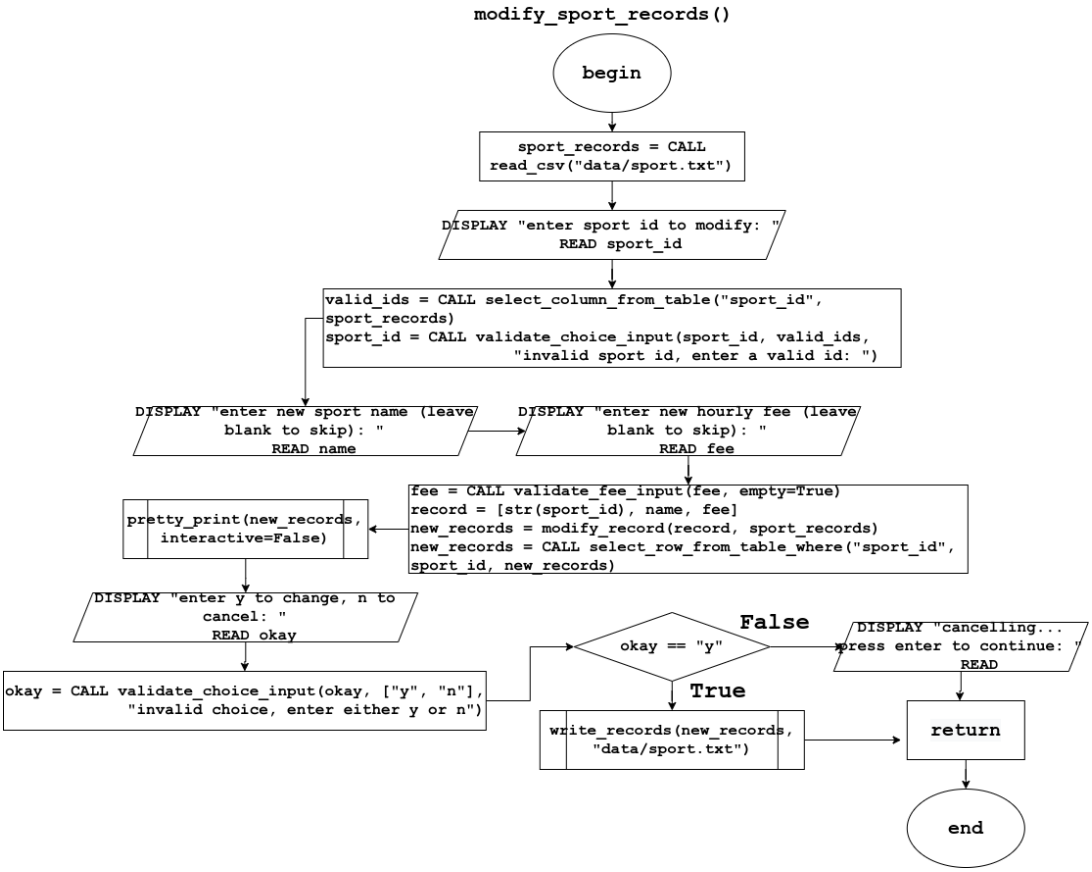
24. Modify_coach_records()



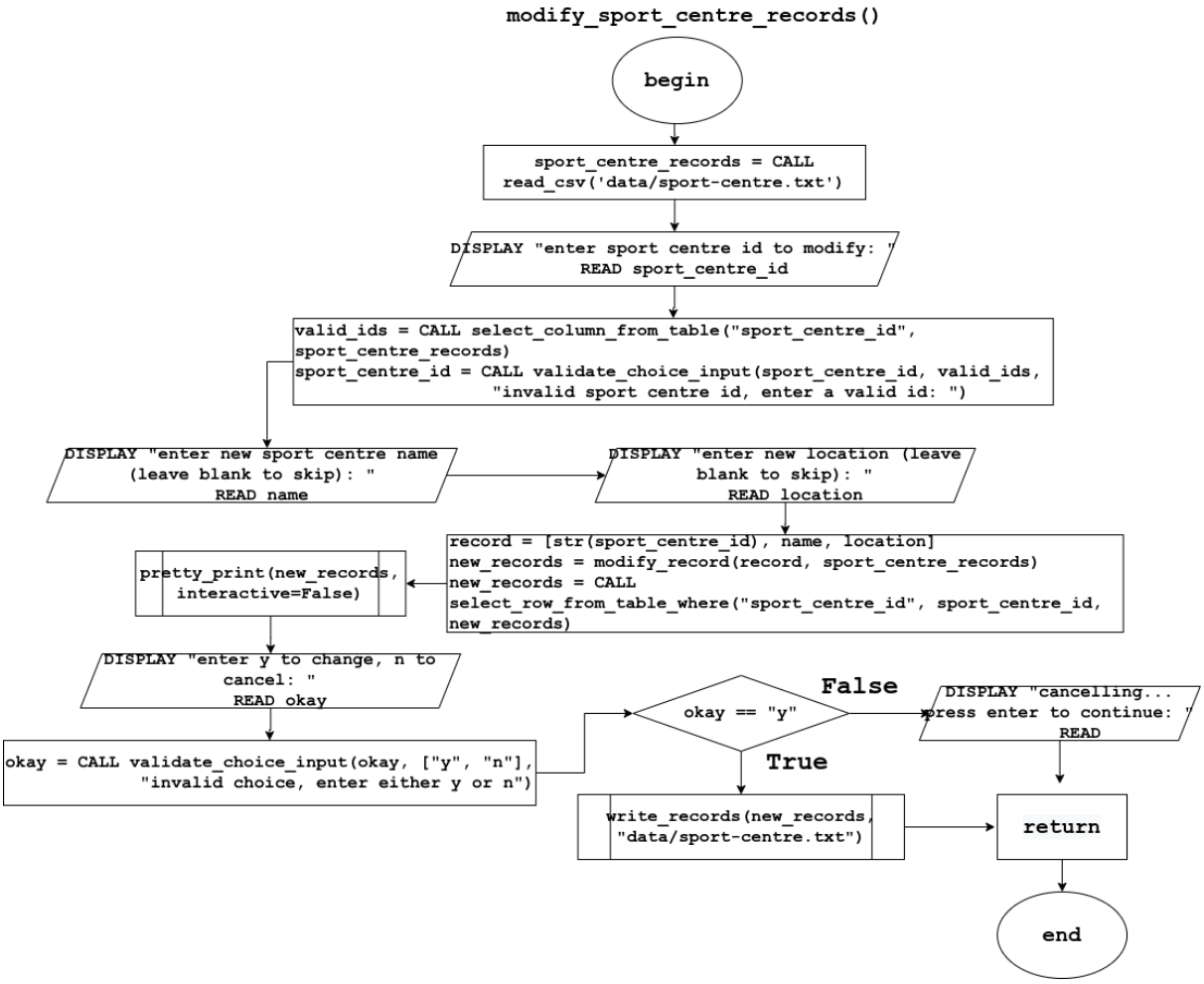
25. Modify_schedule_records()



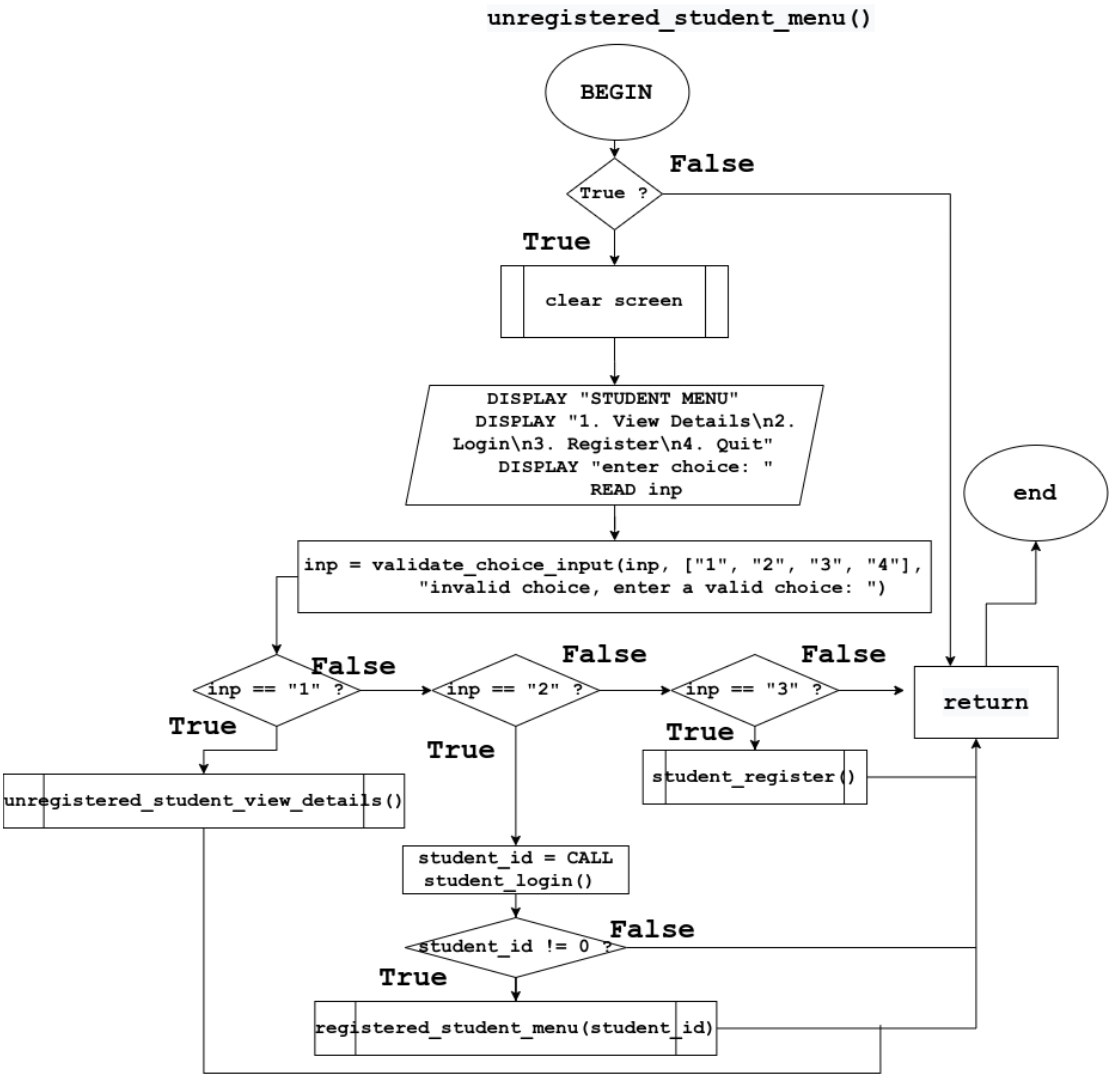
26. Modify_sport_Records()



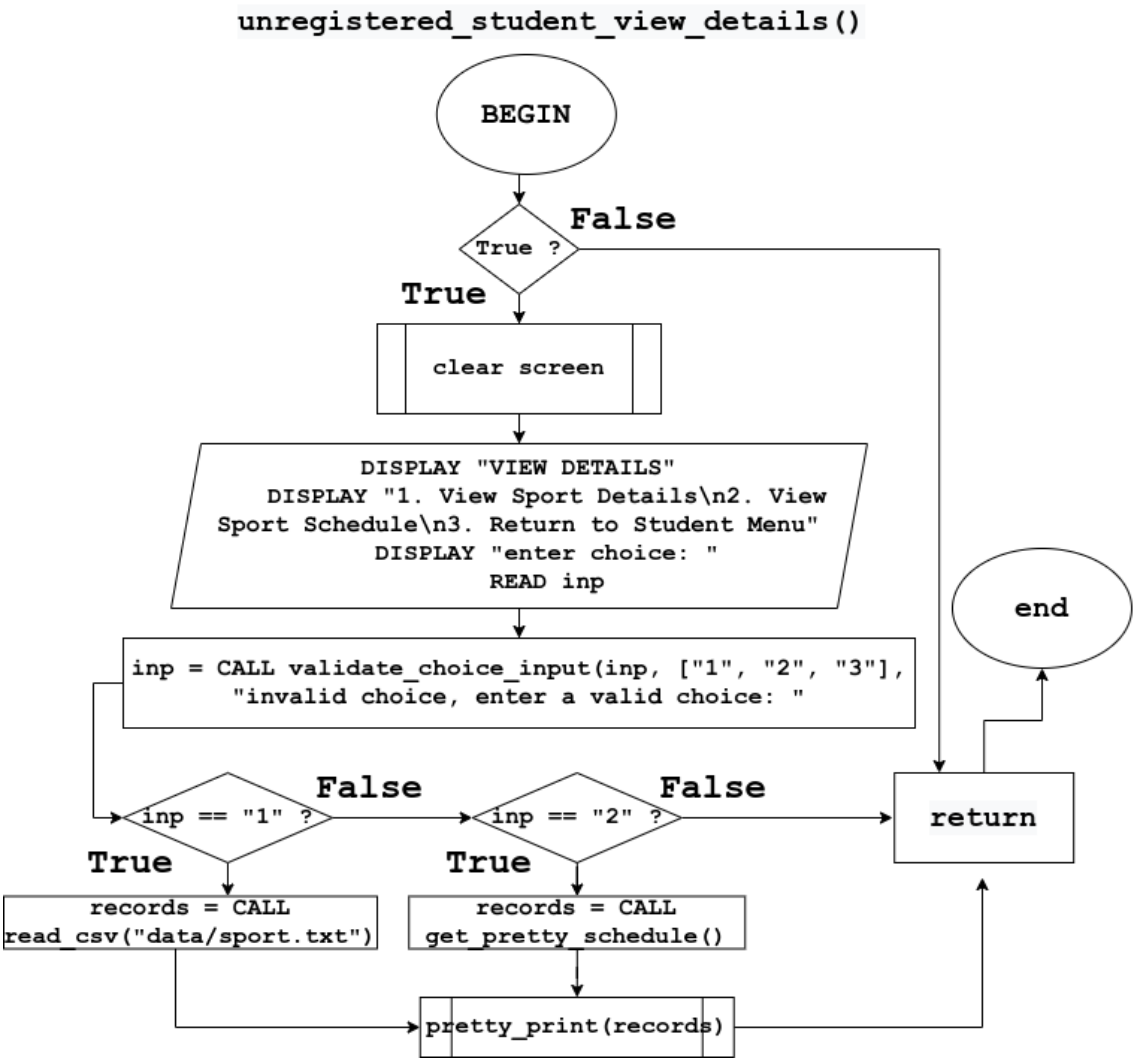
27. Modify_sport_centre_records()



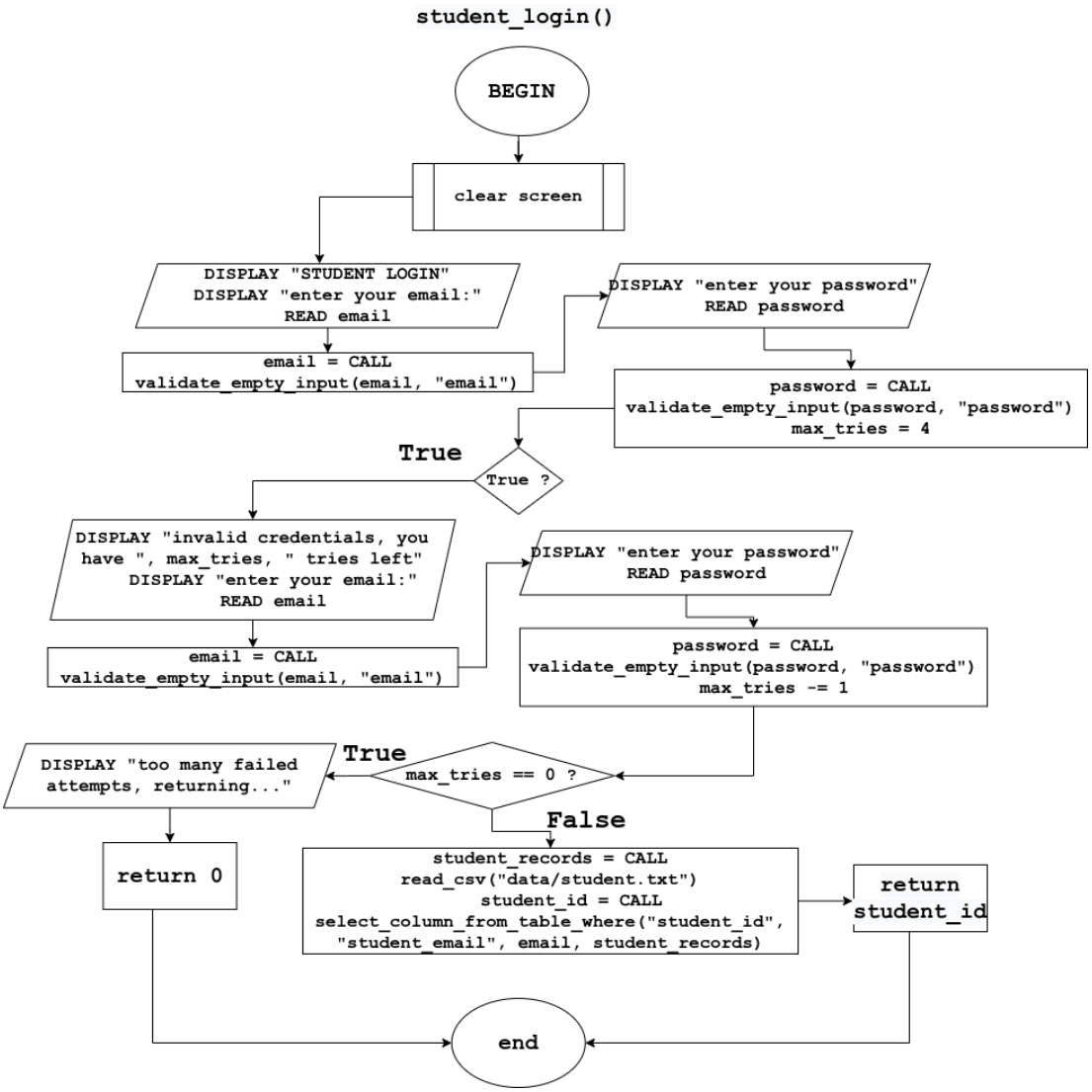
28. Unregistered_student_menu()



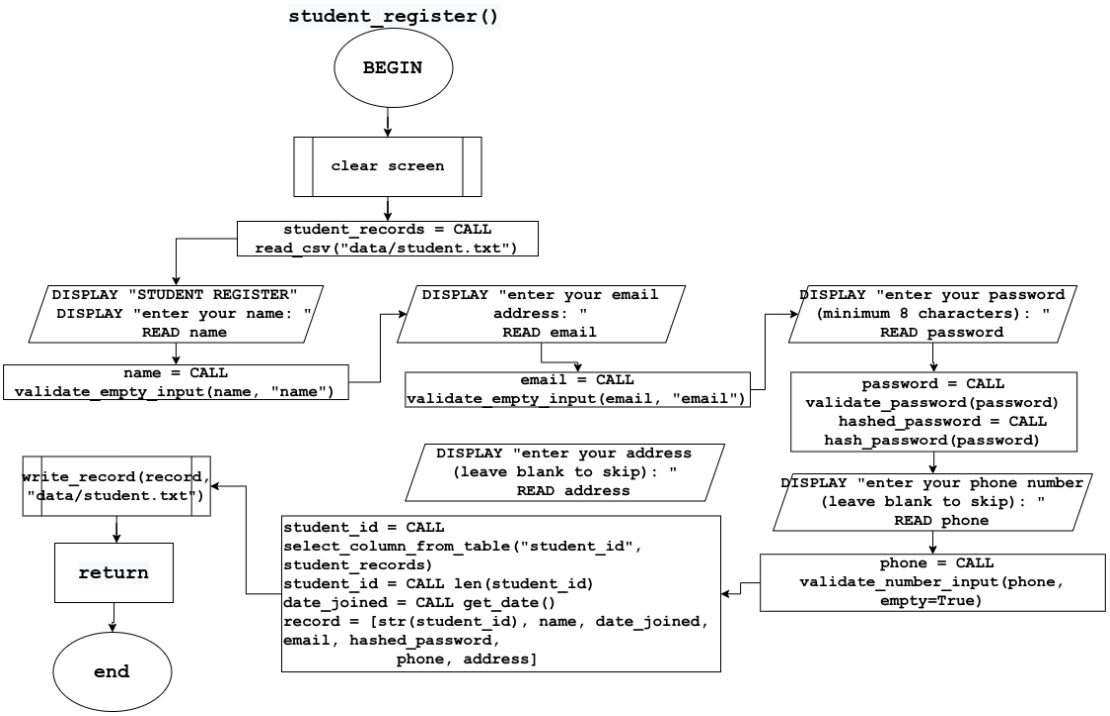
29. Unregistered_student_view_details()



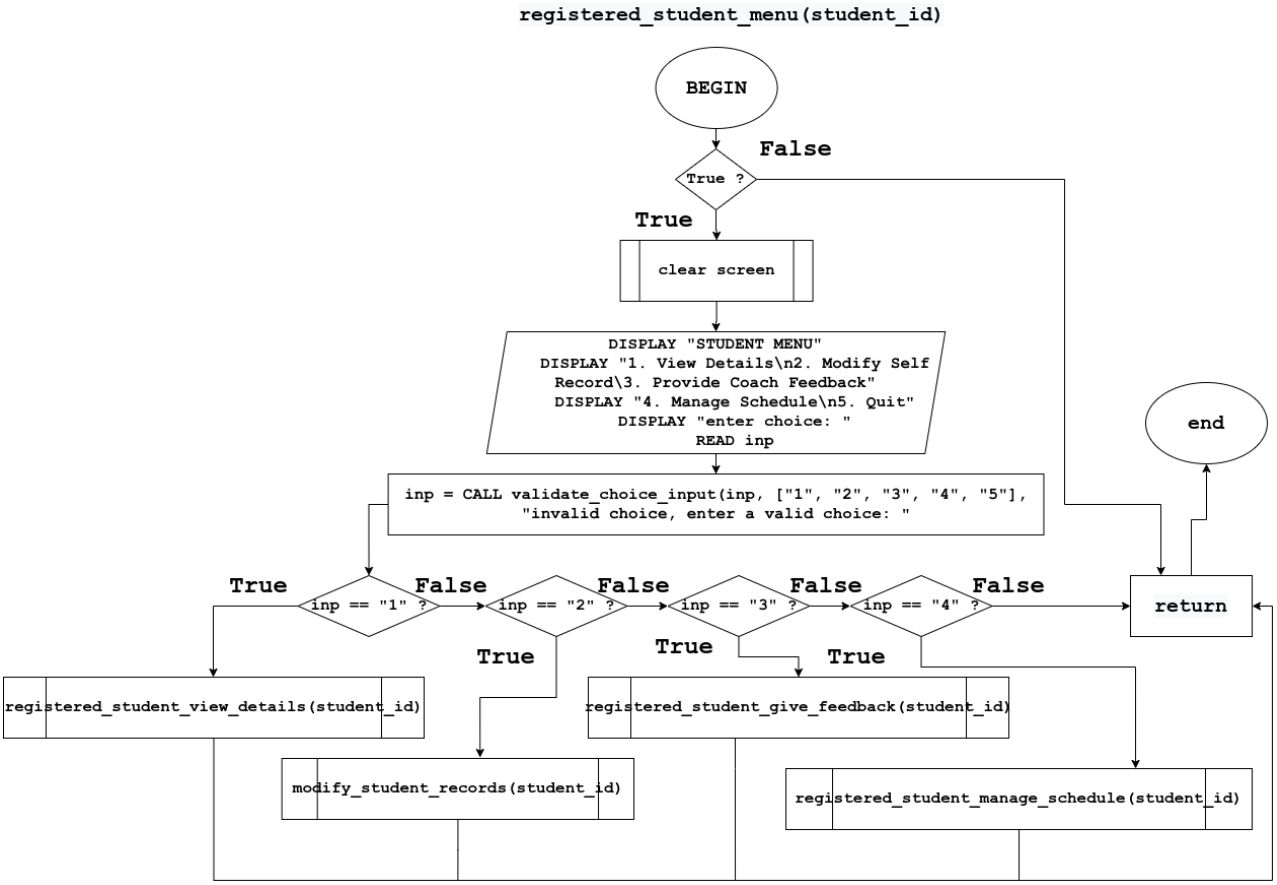
30. Student_login()



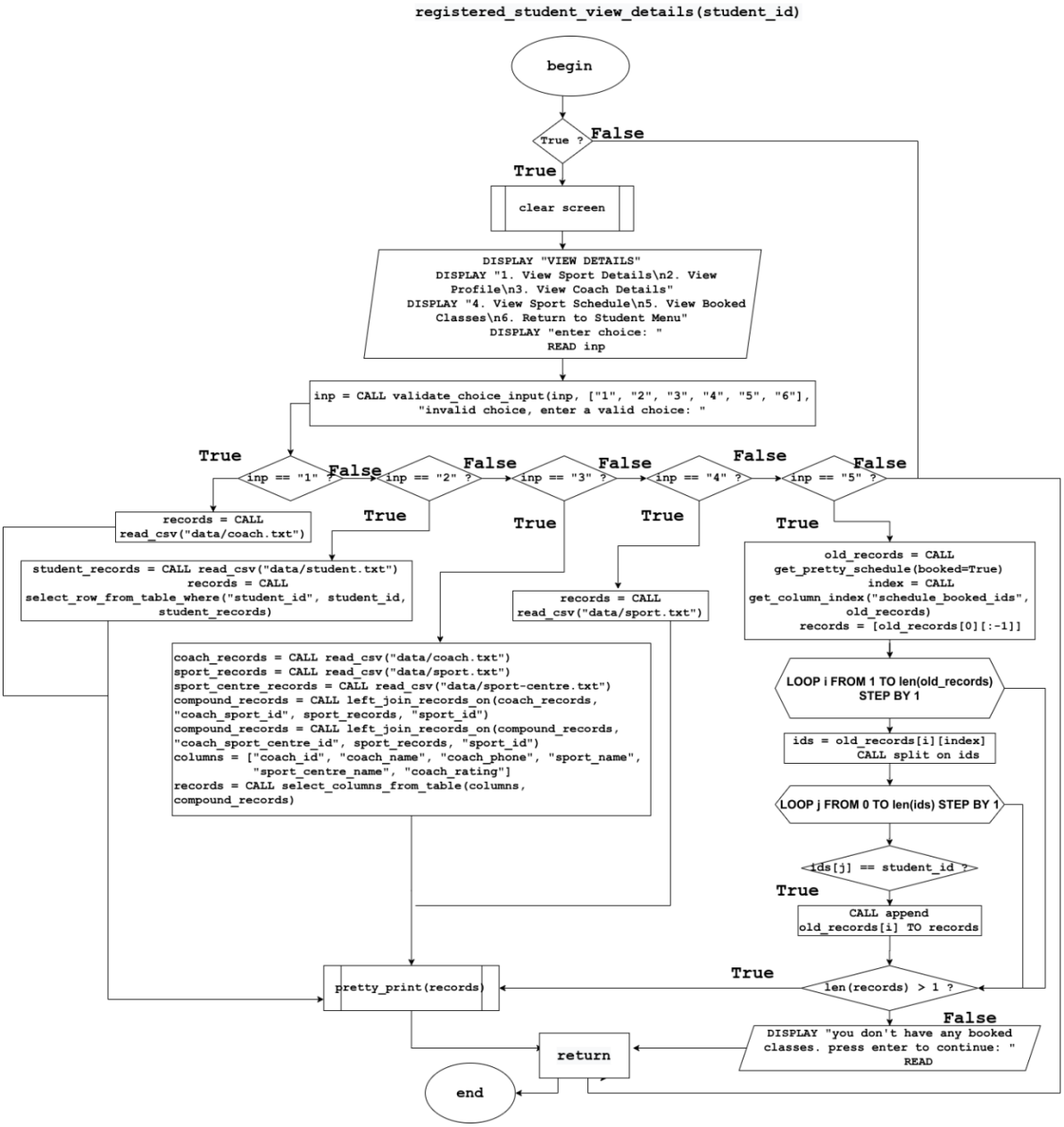
31. Student_register()



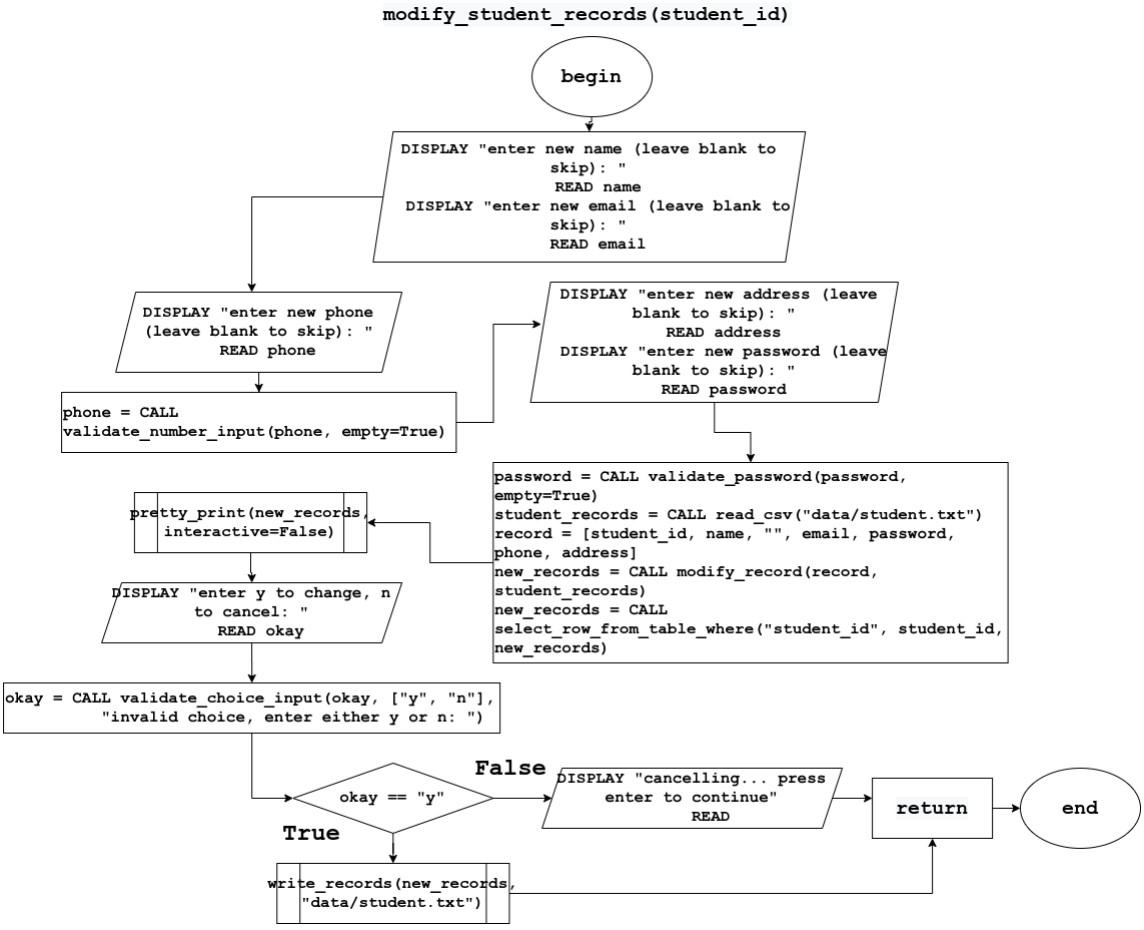
32. Registered_student_menu()



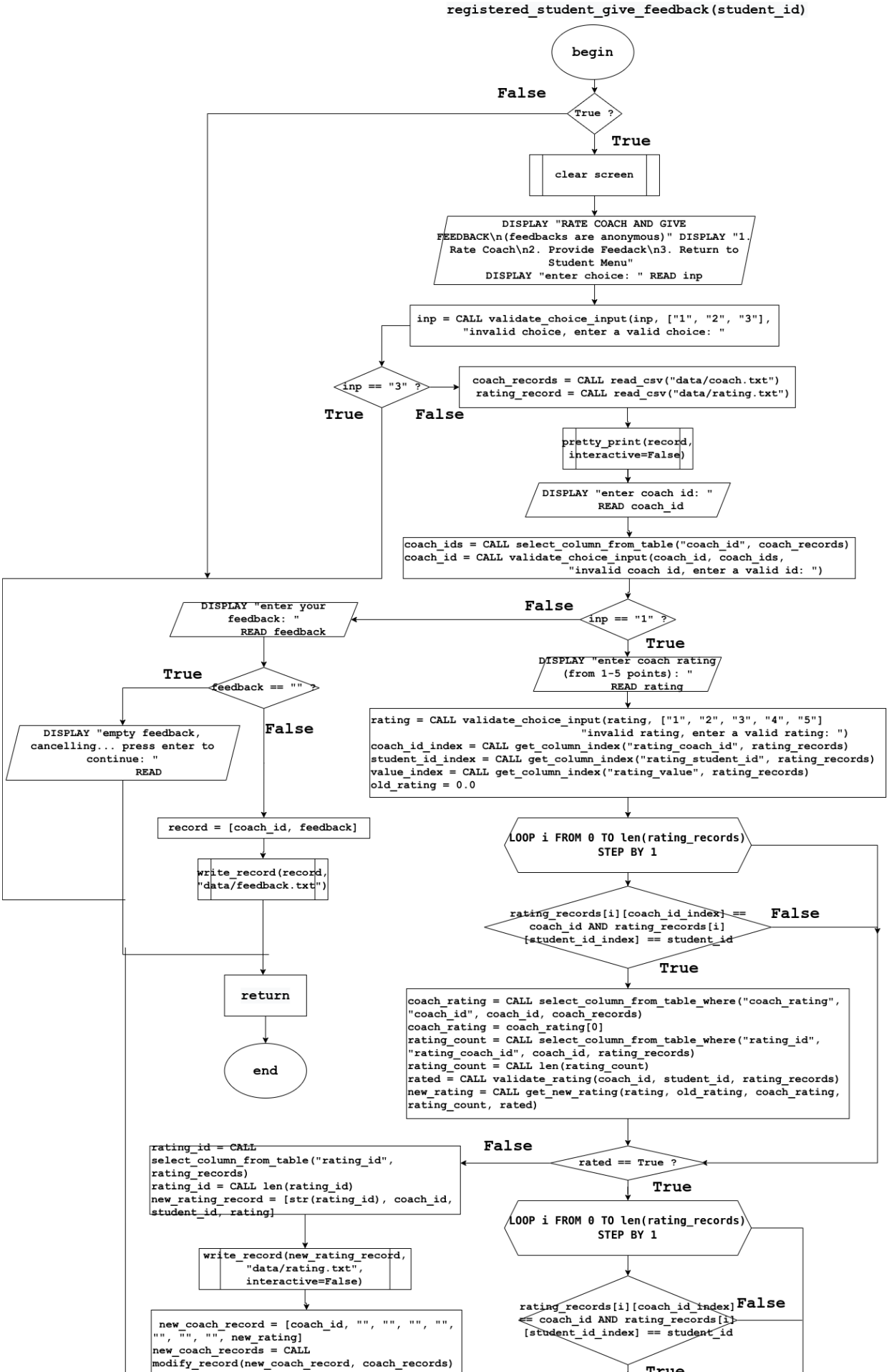
33. Registered_student_view_details()



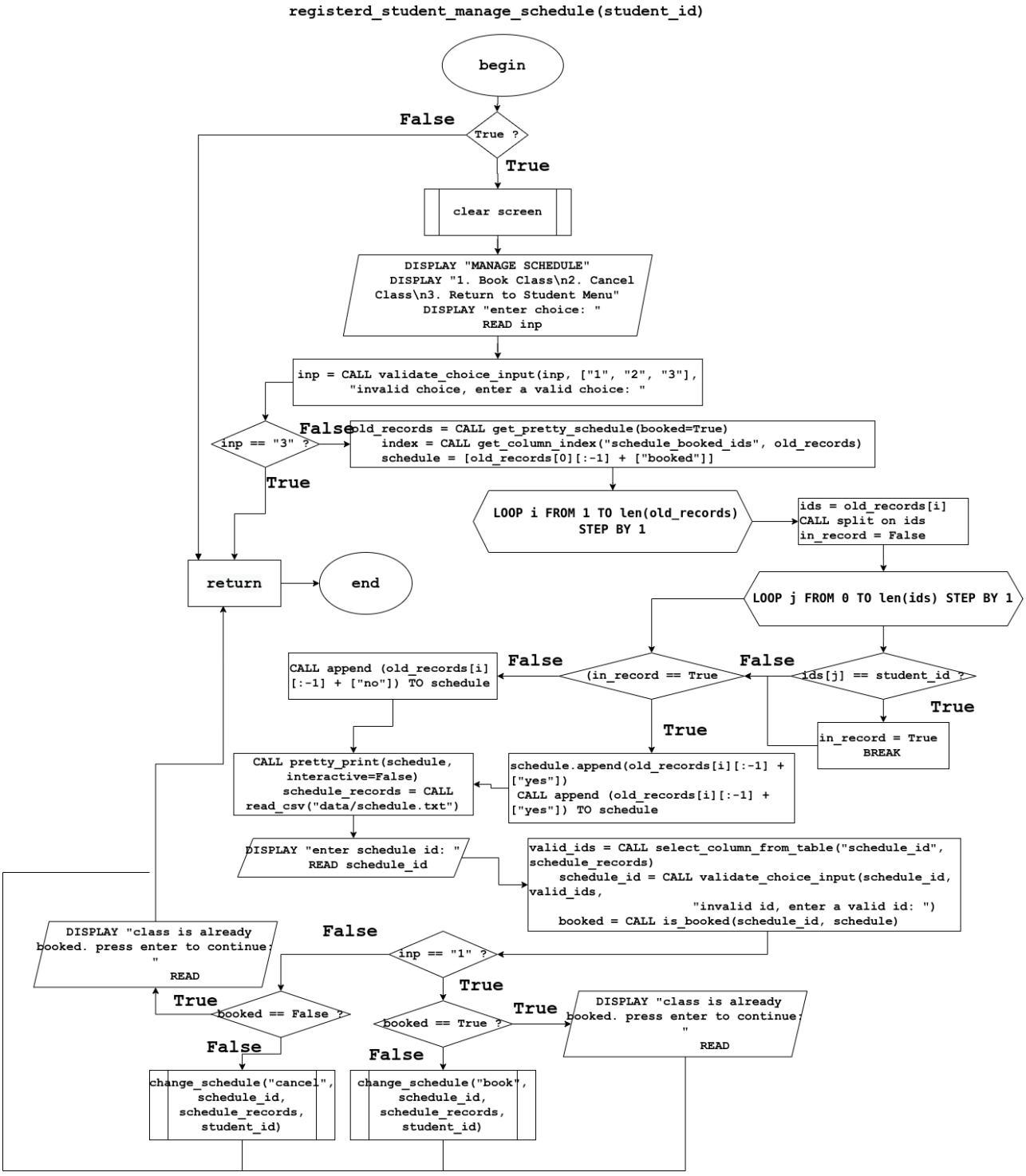
34. Modify_student_records()



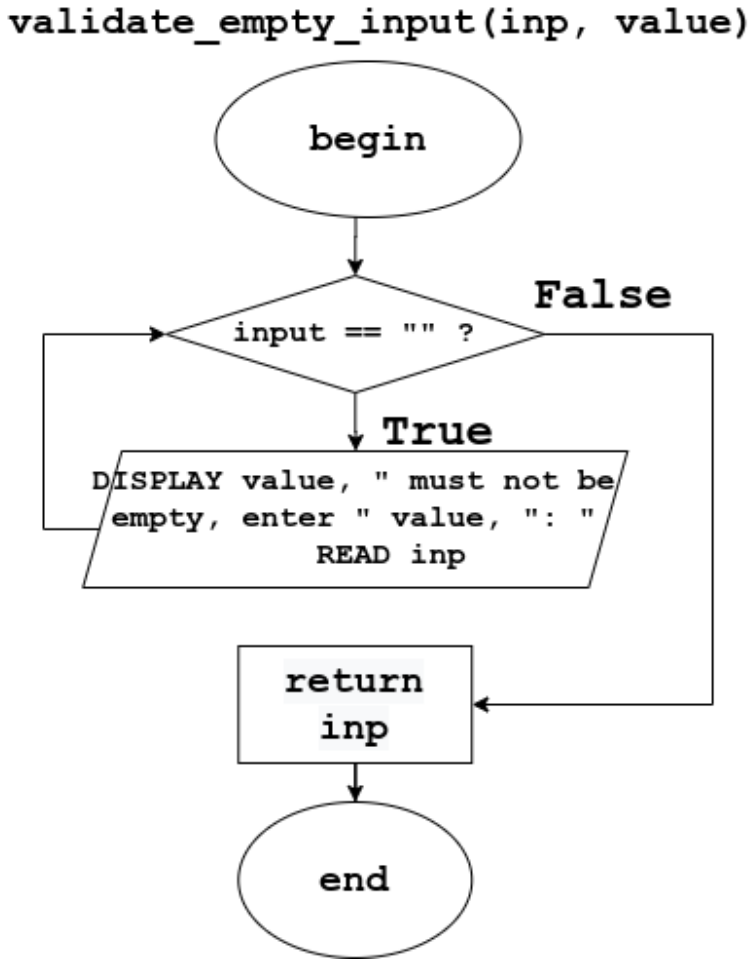
35. Regiestered_students_give_feedback()


registered_student_give_feedback(student_id)
begin
True ?
False
True
clear screen
DISPLAY "RATE COACH AND GIVE FEEDBACK\n(feedbacks are anonymous)" DISPLAY "1. Rate Coach\n2. Provide Feedack\n3. Return to Student Menu" DISPLAY "enter choice: " READ inp
inp = CALL validate_choice_input(inp, ["1", "2", "3"], "invalid choice, enter a valid choice: "
inp == "3" ?
True
False
coach_records = CALL read_csv("data/coach.txt")
rating_record = CALL read_csv("data/rating.txt")
pretty_print(record, interactive=False)
DISPLAY "enter coach id: " READ coach_id
coach_ids = CALL select_column_from_table("coach_id", coach_records)
coach_id = CALL validate_choice_input(coach_id, coach_ids, "invalid coach id, enter a valid id: ")
inp == "1" ?
False
True
DISPLAY "enter your feedback: " READ feedback
DISPLAY "enter coach rating (from 1-5 points): " READ rating
feedback == "" ?
True
False
DISPLAY "empty feedback, cancelling... press enter to continue: " READ
rating = CALL validate_choice_input(rating, ["1", "2", "3", "4", "5"] "invalid rating, enter a valid rating: ")
coach_id_index = CALL get_column_index("rating_coach_id", rating_records)
student_id_index = CALL get_column_index("rating_student_id", rating_records)
value_index = CALL get_column_index("rating_value", rating_records)
old_rating = 0.0
record = [coach_id, feedback]
write_record(record, "data/feedback.txt")
LOOP i FROM 0 TO len(rating_records) STEP BY 1
rating_records[i][coach_id_index] == coach_id AND rating_records[i][student_id_index] == student_id
False
True
return
end
coach_rating = CALL select_column_from_table_where("coach_rating", "coach_id", coach_id, coach_records)
coach_rating = coach_rating[0]
rating_count = CALL select_column_from_table_where("rating_id", "rating_coach_id", coach_id, rating_records)
rating_count = CALL len(rating_count)
rated = CALL validate_rating(coach_id, student_id, rating_records)
new_rating = CALL get_new_rating(rating, old_rating, coach_rating, rating_count, rated)
rated == True ?
False
True
rating_id = CALL select_column_from_table("rating_id", rating_records)
rating_id = CALL len(rating_id)
new_rating_record = [str(rating_id), coach_id, student_id, rating]
write_record(new_rating_record, "data/rating.txt", interactive=False)
new_coach_record = [coach_id, "", "", "", "", "", "", "", new_rating]
new_coach_records = CALL modify_record(new_coach_record, coach_records)
LOOP i FROM 0 TO len(rating_records) STEP BY 1
rating_records[i][coach_id_index] == coach_id AND rating_records[i][student_id_index] == student_id
False
True

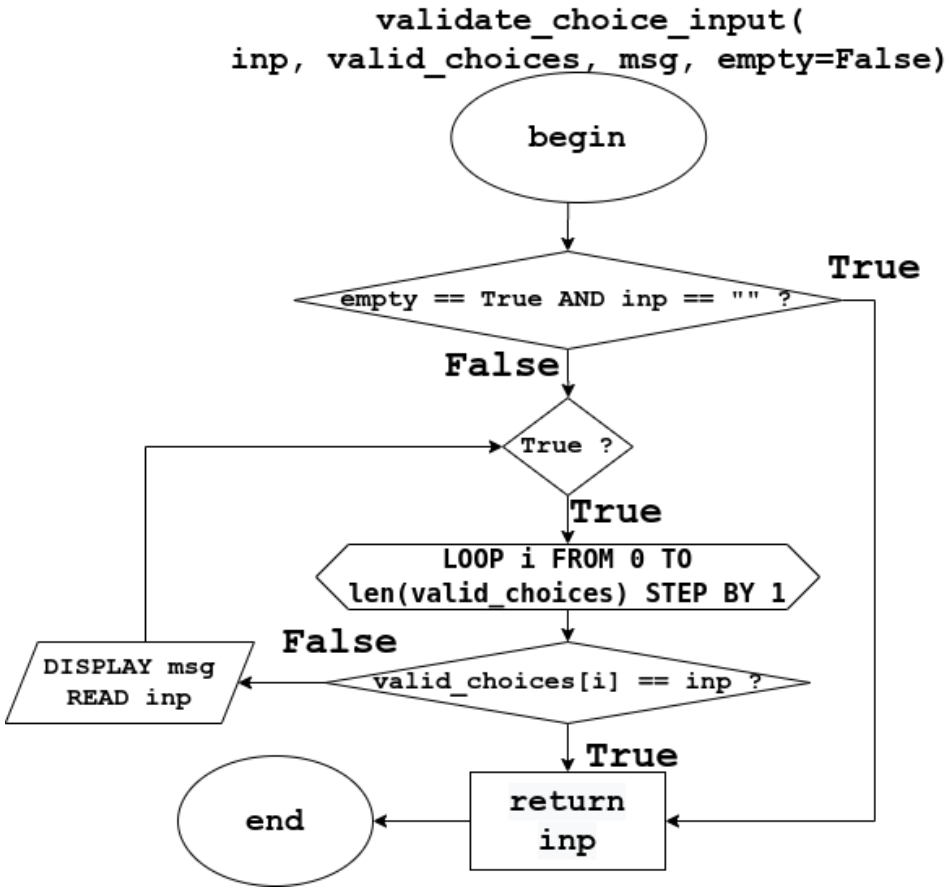
36. Registered_students_manage_schedule()



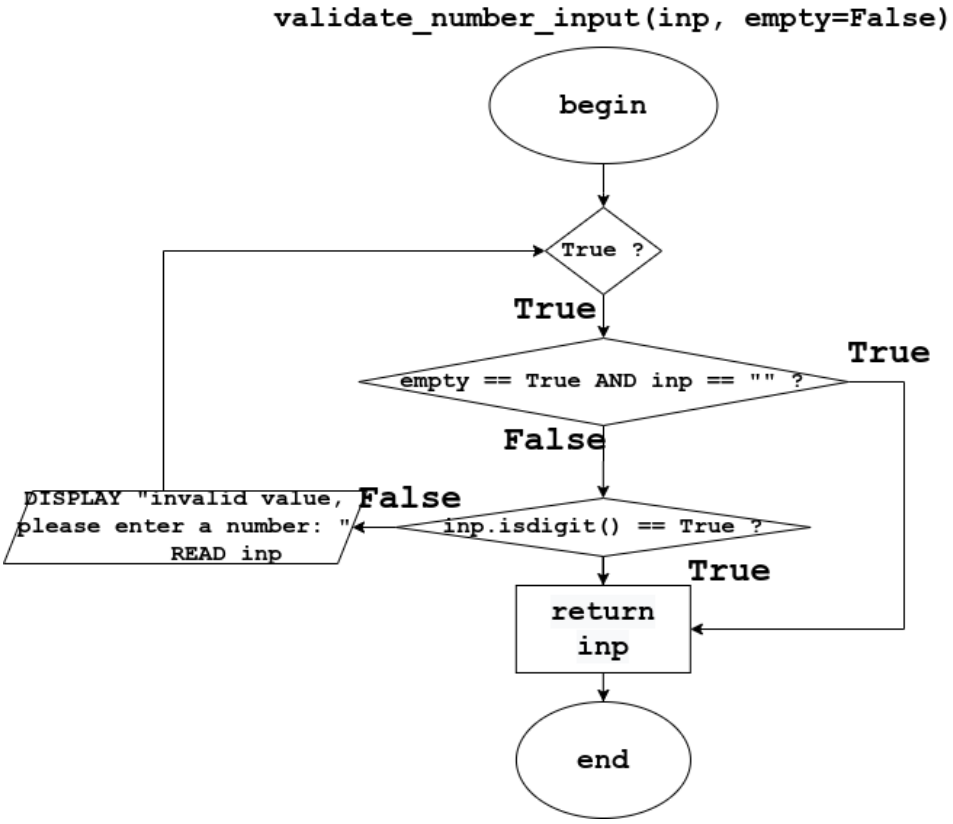
37. Validate_empty_input()



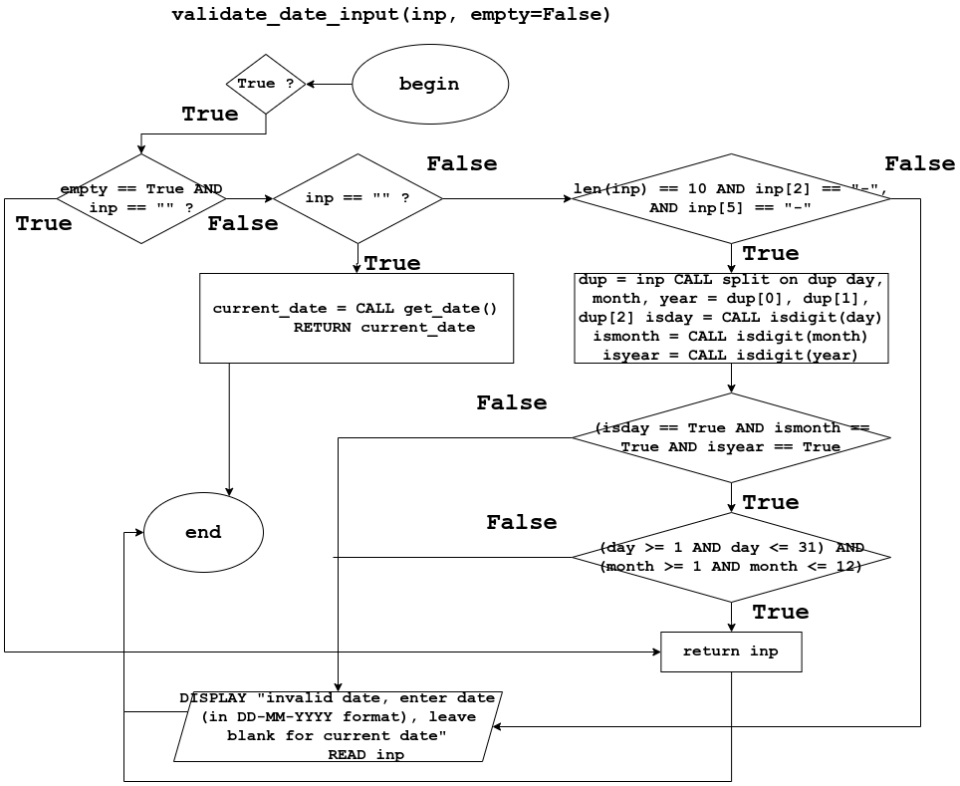
38. Validate_choice_input()



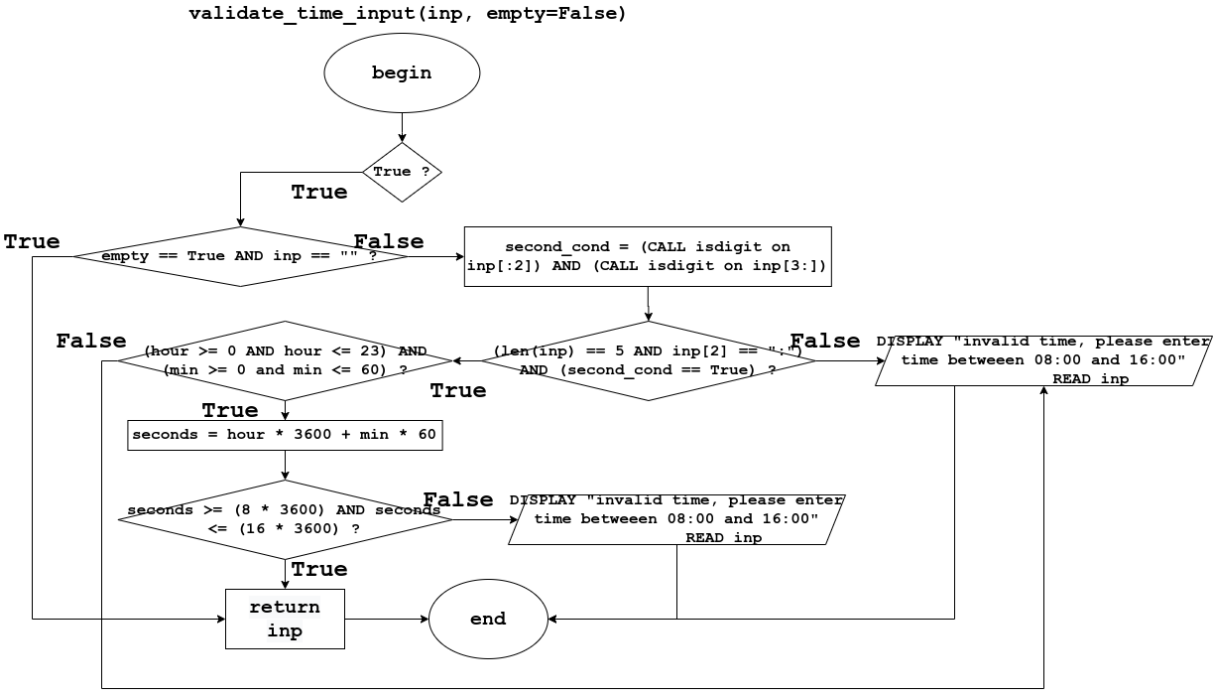
39. Validate_number_input()



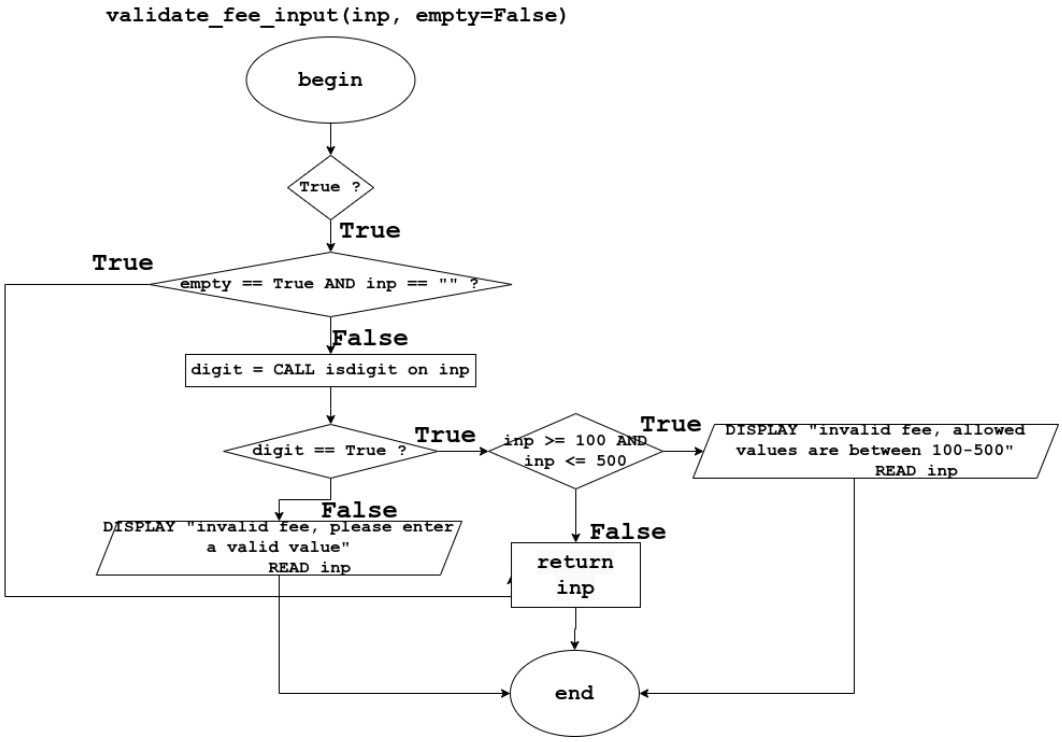
40. Validate_date_input()



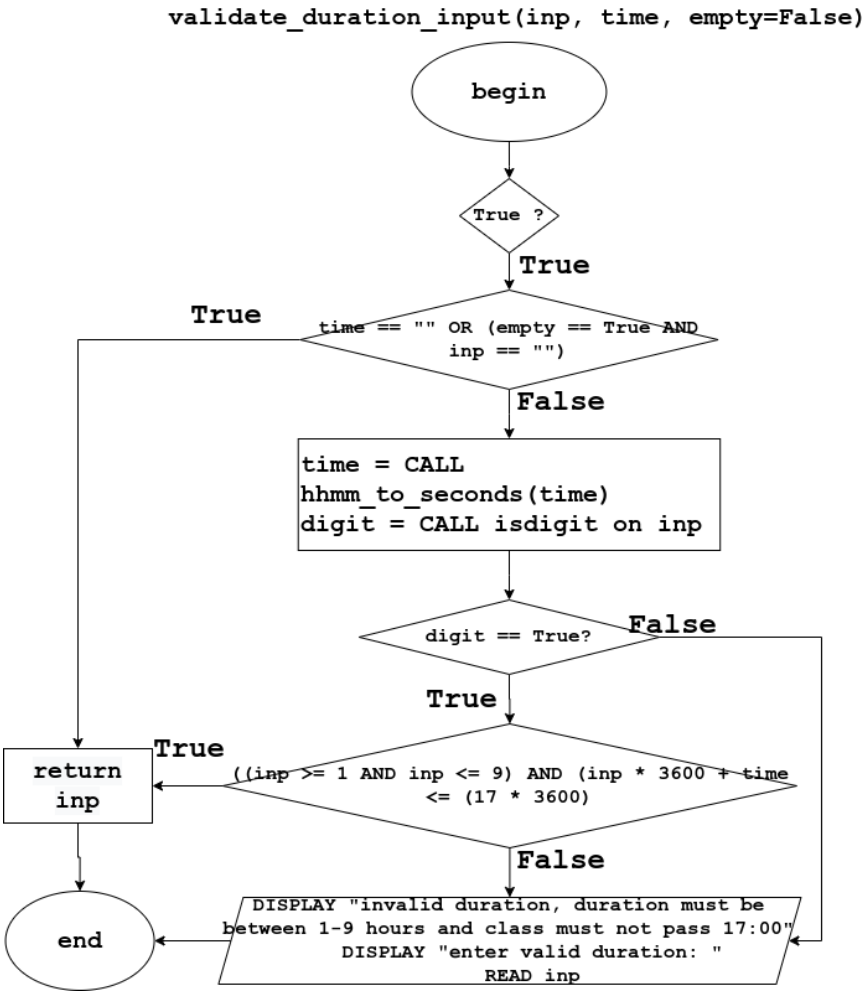
41. Validate_time_input()



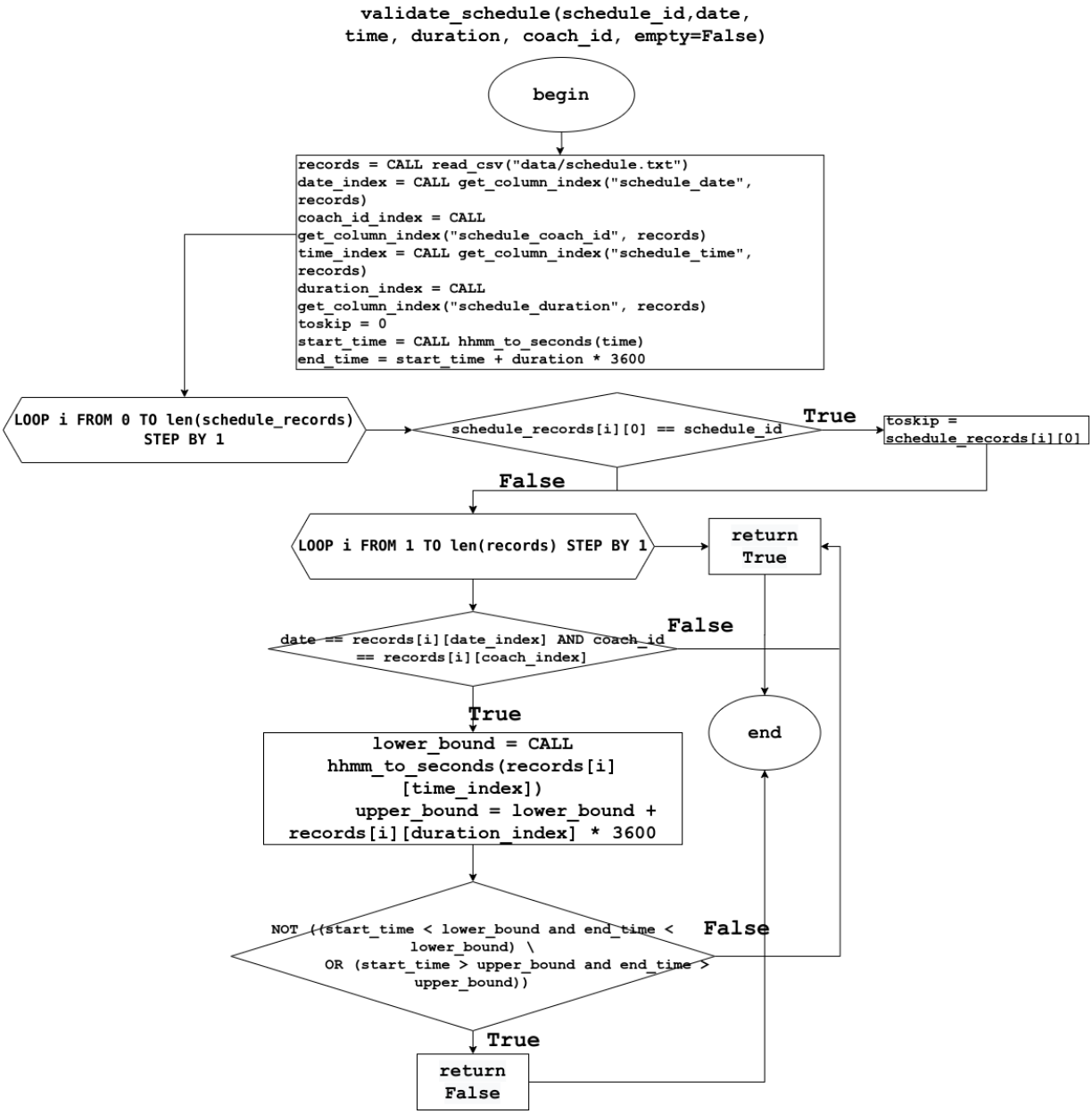
42. Validate_fee_input()



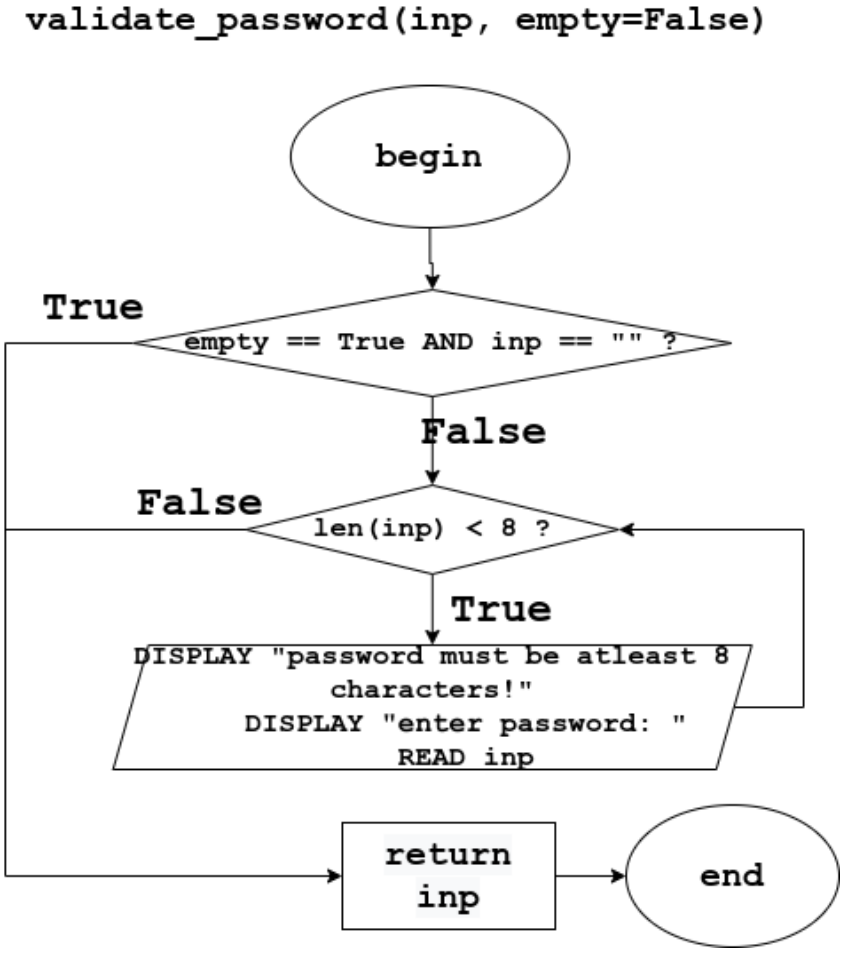
43. Validate_duration_input()



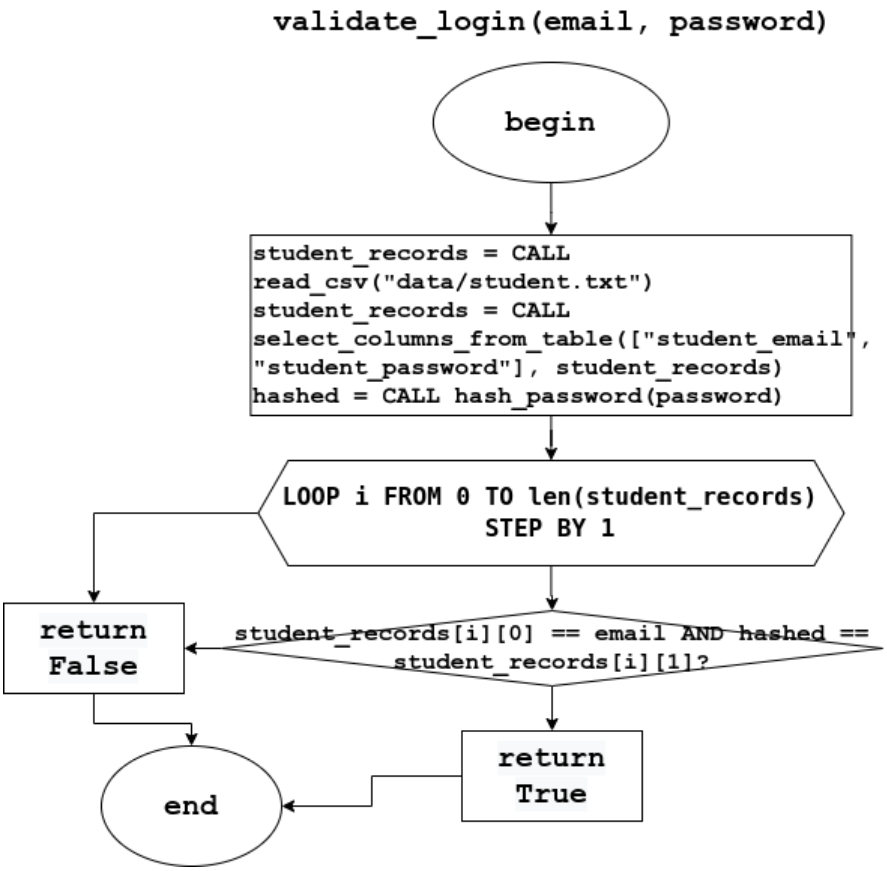
44. Validate_schedule()



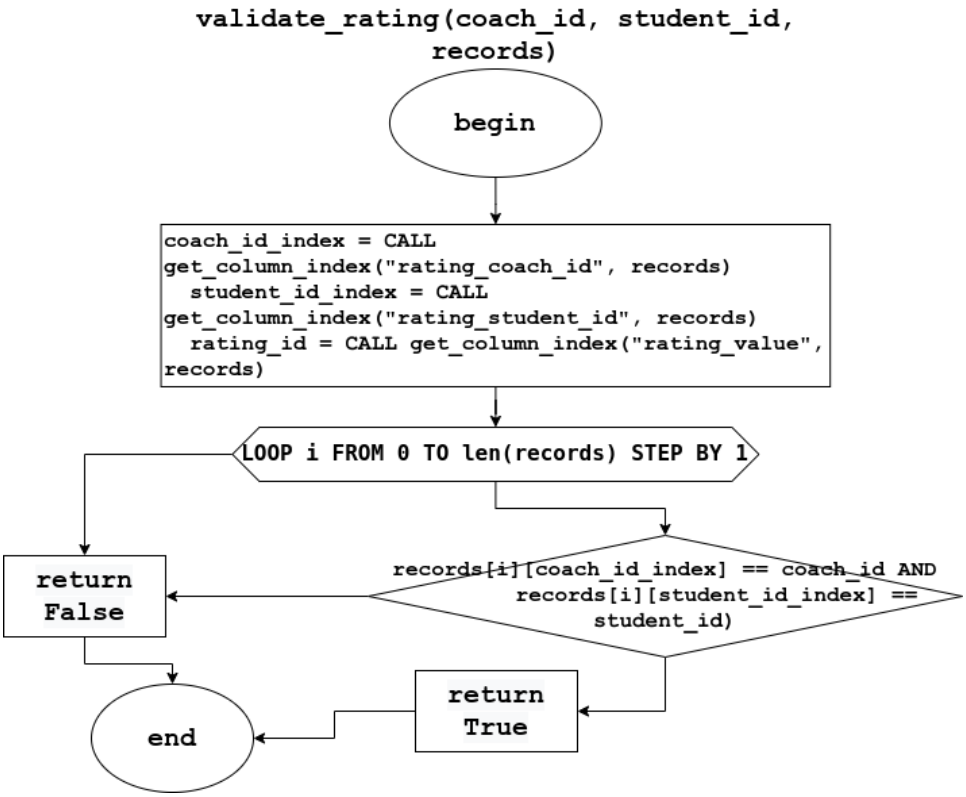
45. Validate_password()



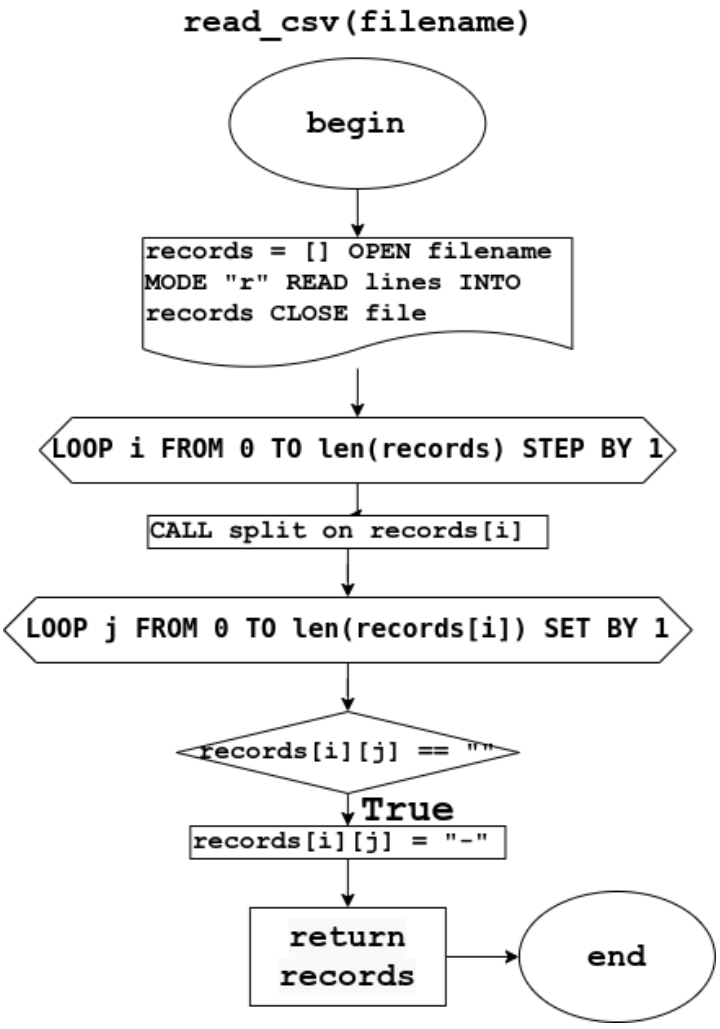
46. Validate_login()



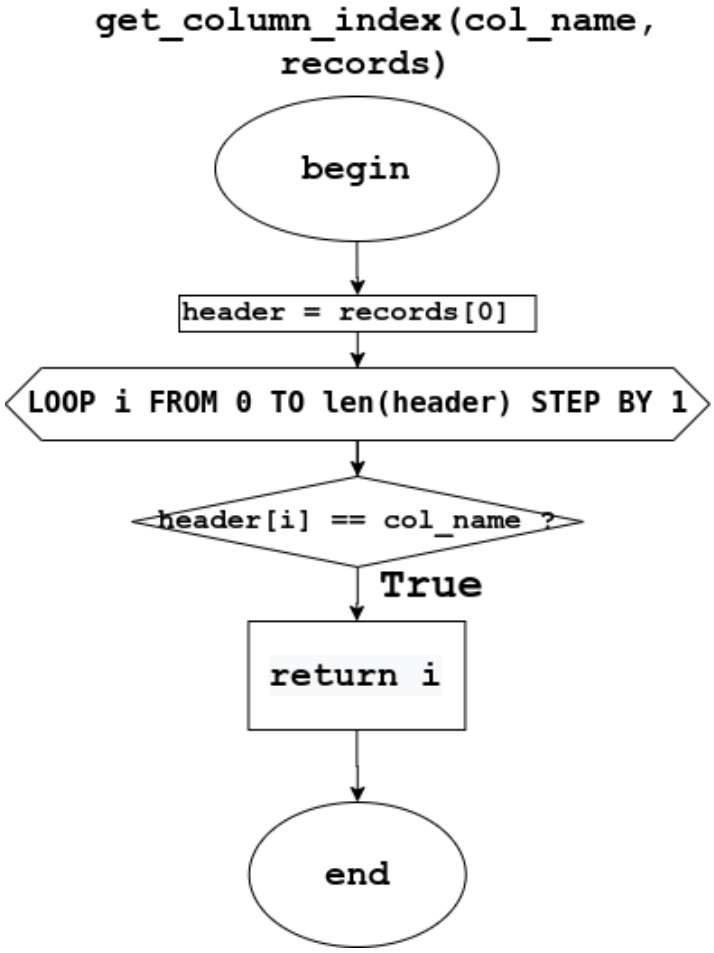
47. Validate_rating()



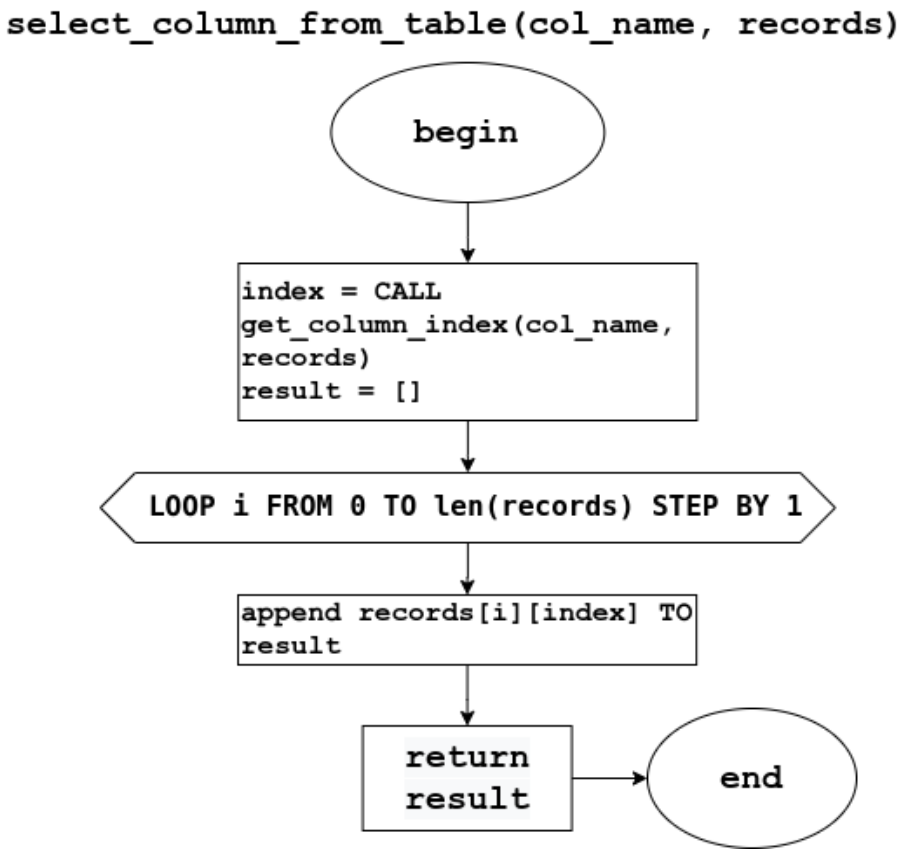
48. Read_csv()


read_csv(filename)
begin
records = [] OPEN filename MODE "r" READ lines INTO records CLOSE file
LOOP i FROM 0 TO len(records) STEP BY 1
CALL split on records[i]
LOOP j FROM 0 TO len(records[i]) SET BY 1
records[i][j] == ""
True
records[i][j] = "-"
return records
end

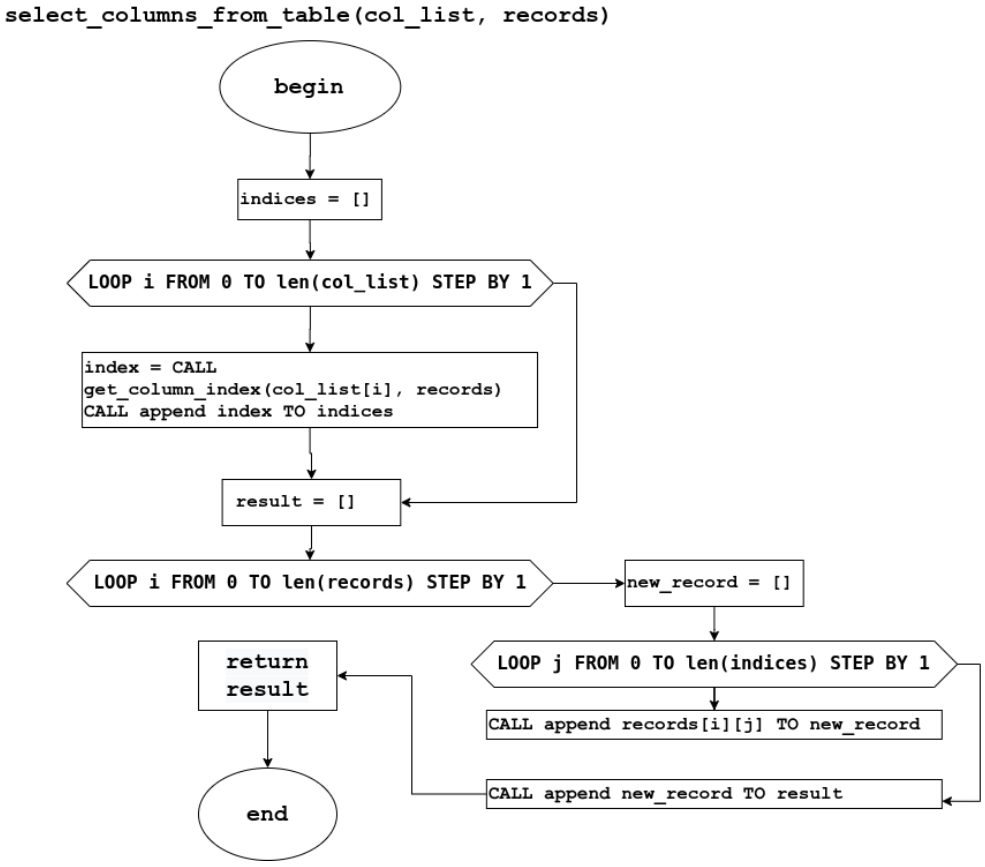
49. Get_column_index()



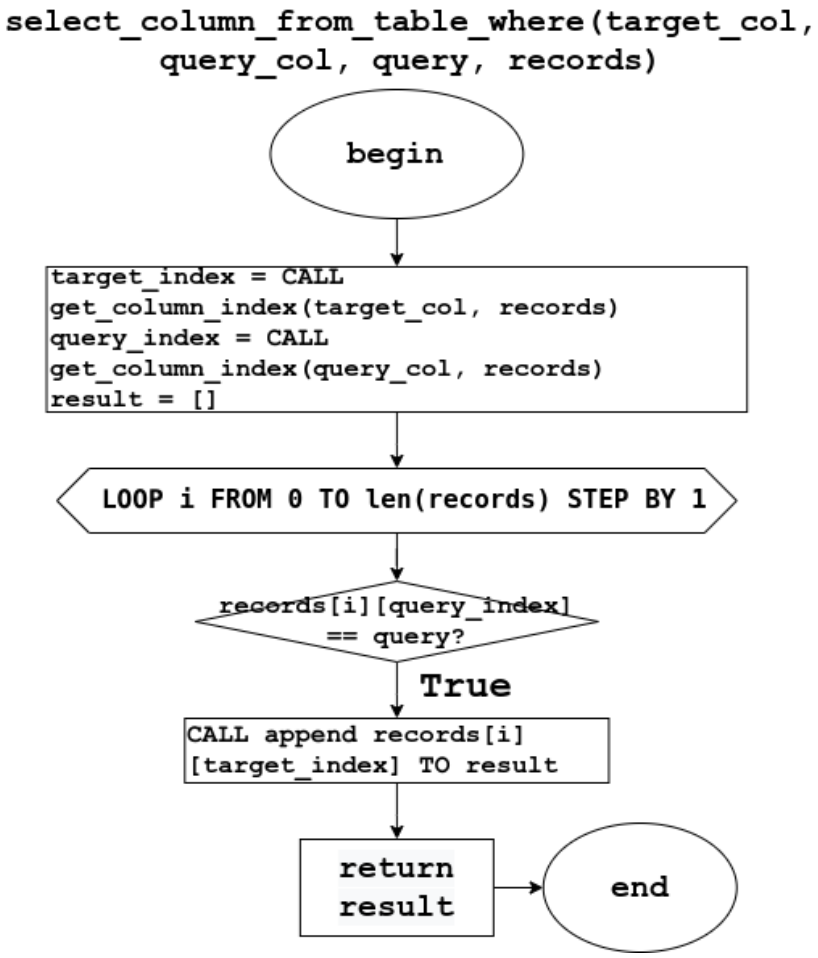
50. Select_column_from_table()



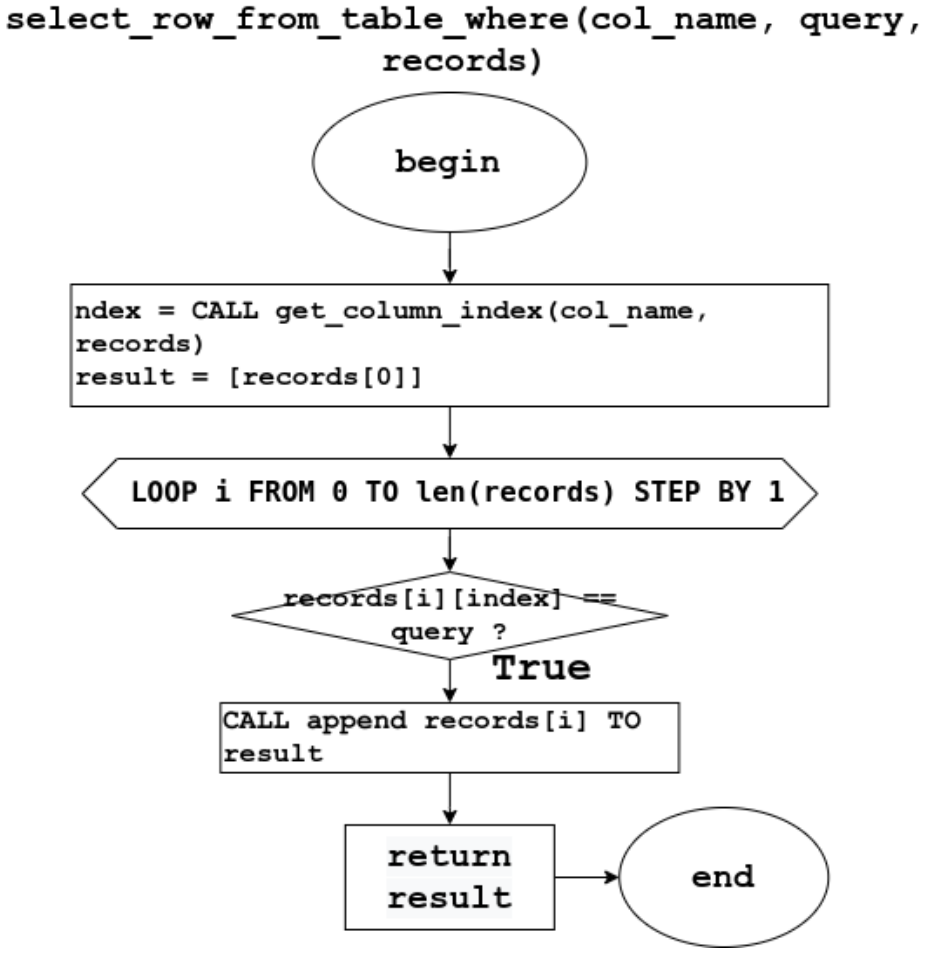
51. Select_columns_from_table()



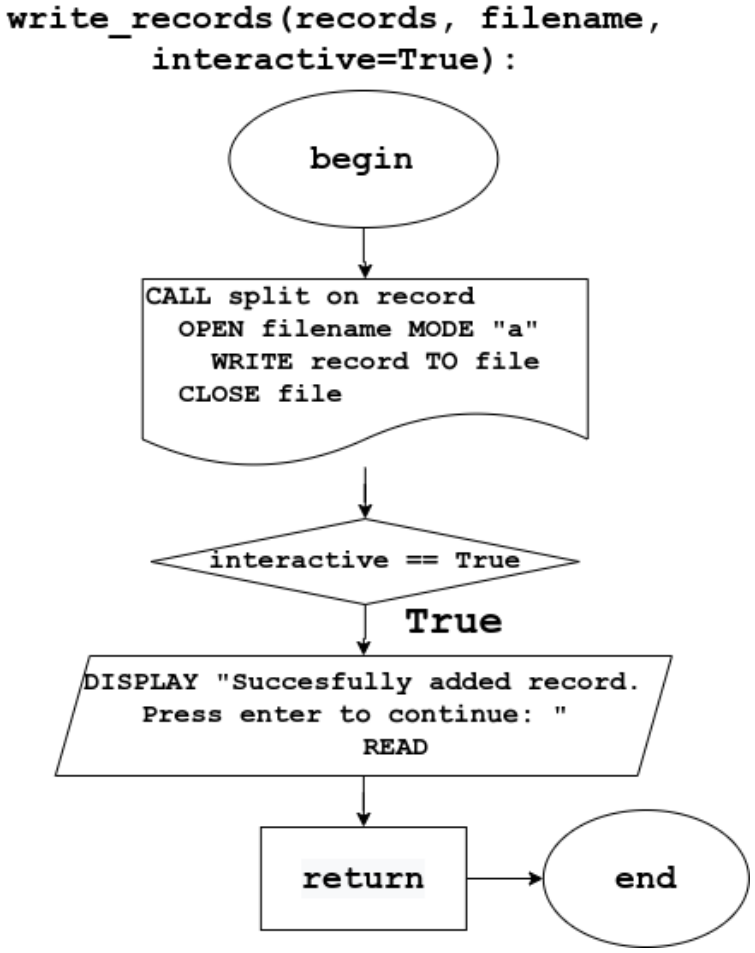
52. Select_column_from_table_where()



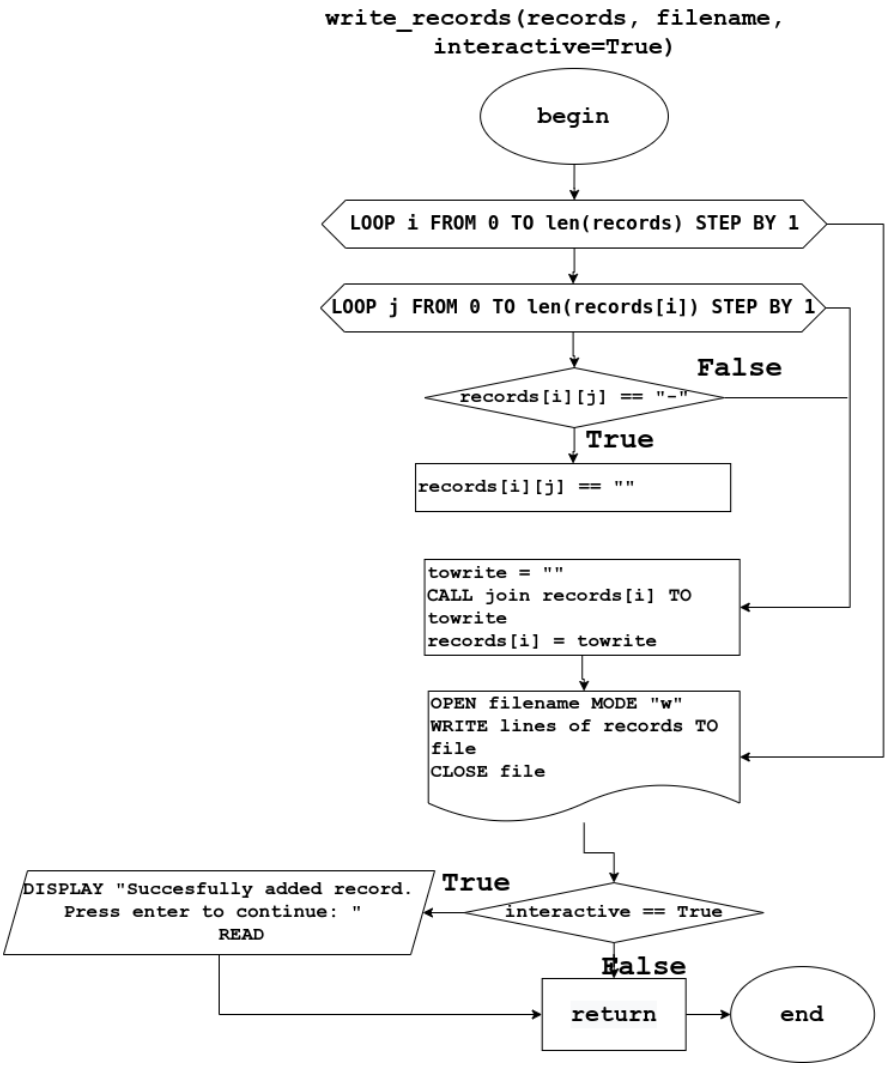
53. Select_row_from_table_where()



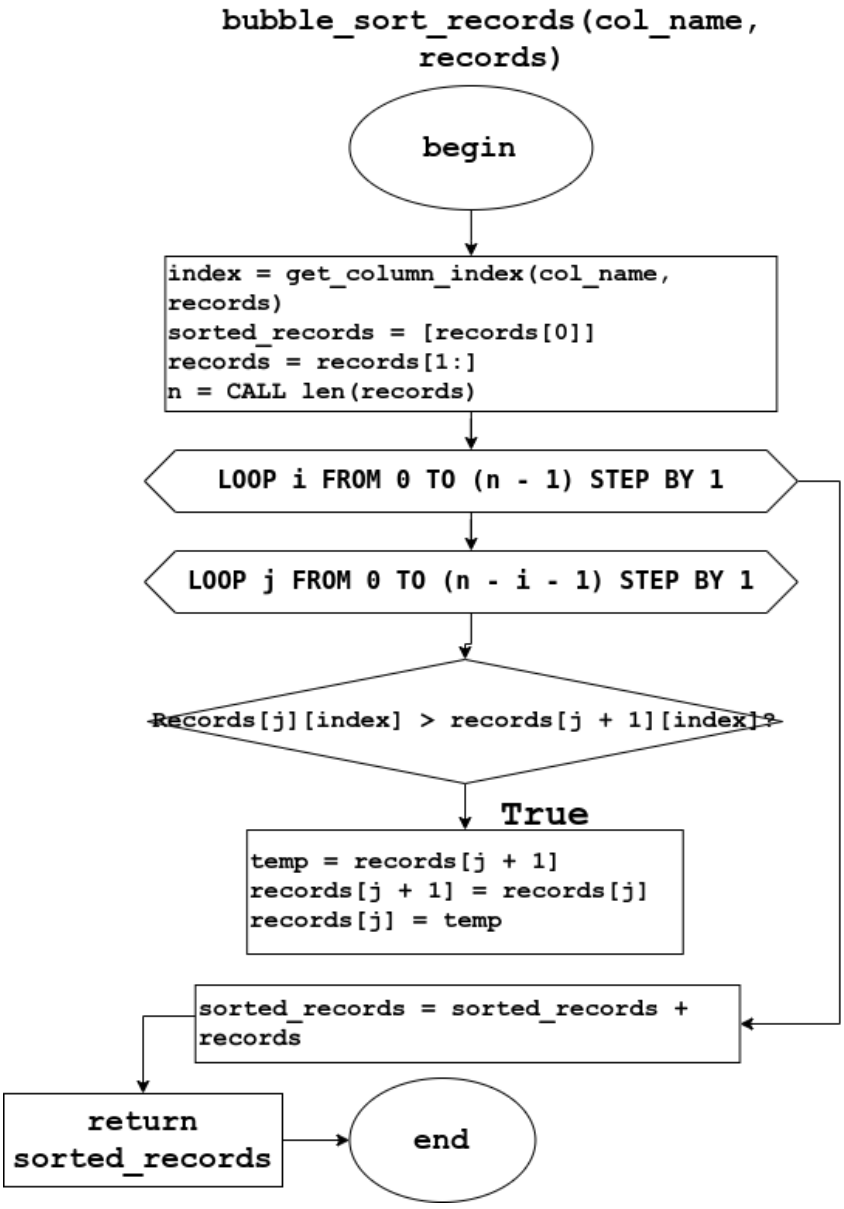
54. Write_record()



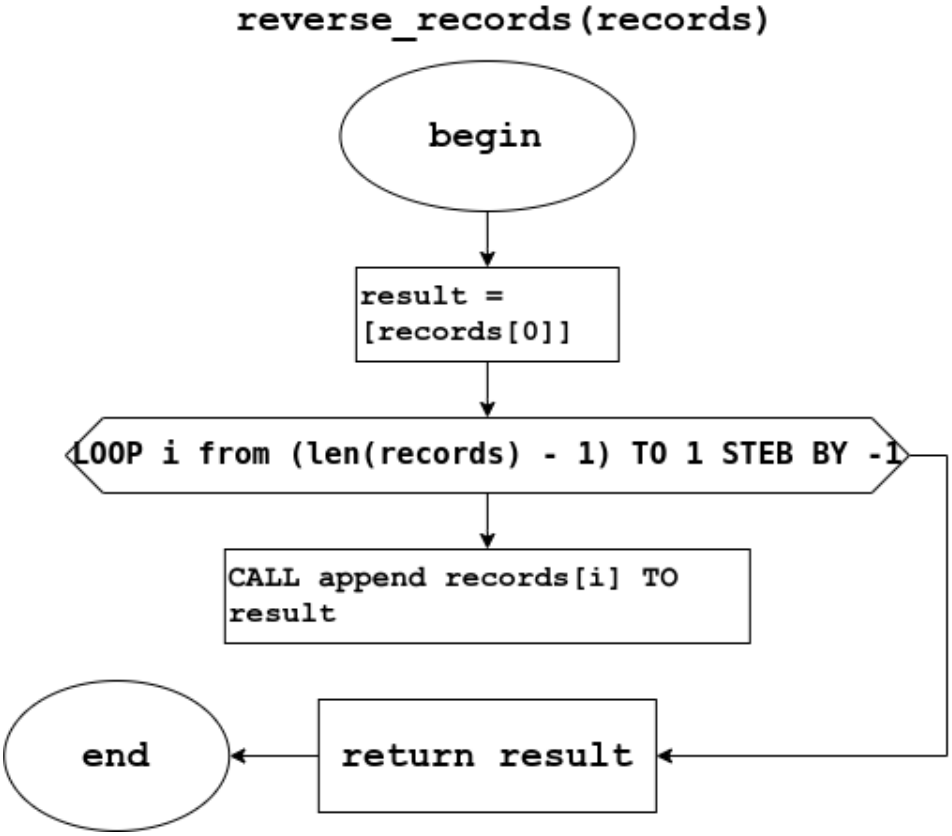
55. Write_records()



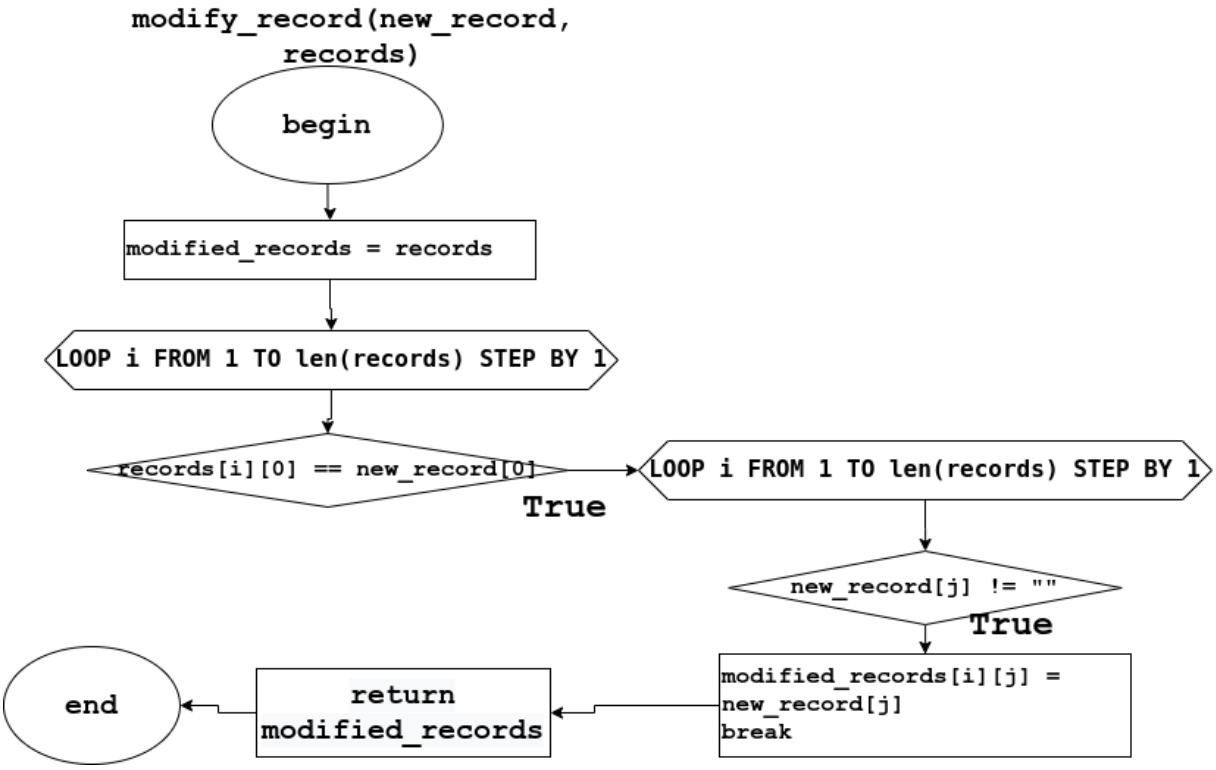
56. Bubble_sort_records()



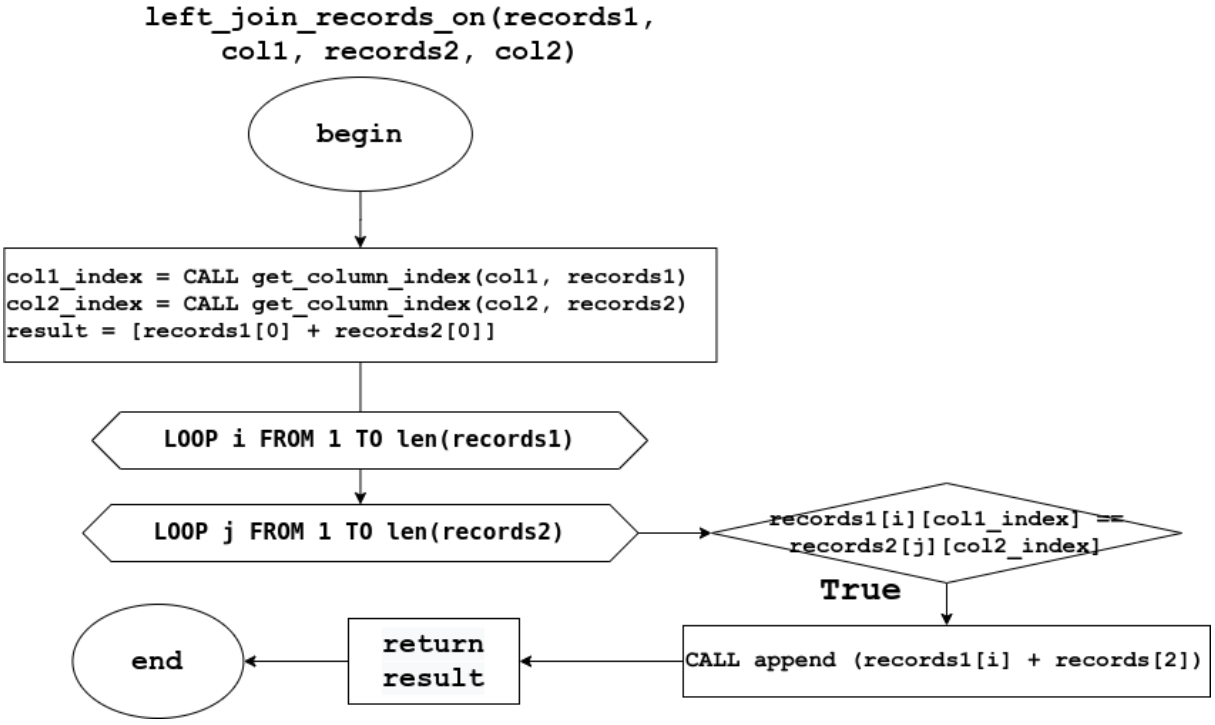
57. Reverse_records()



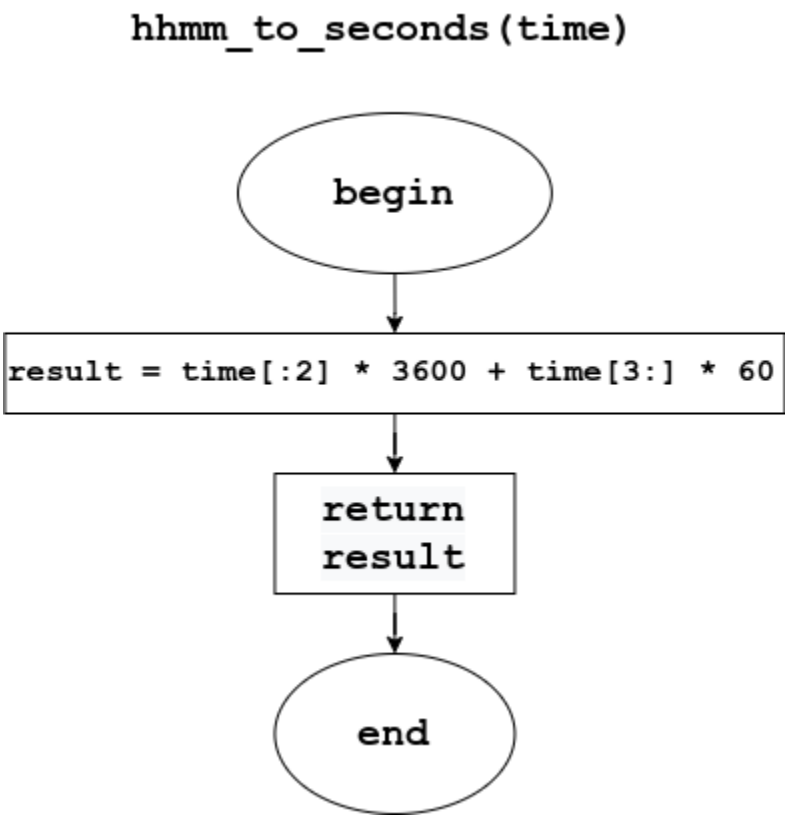
58. Modify_records()



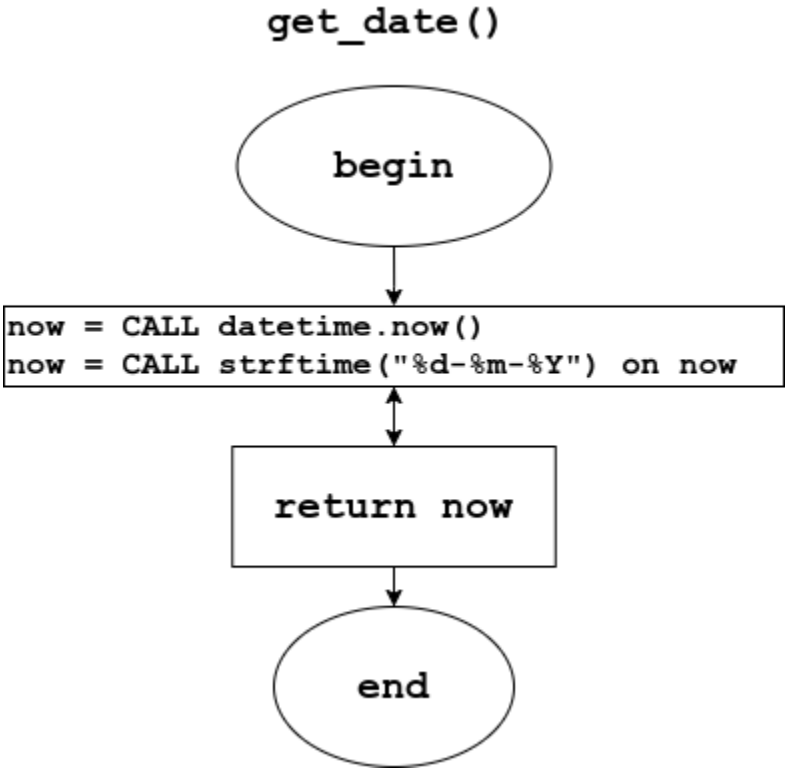
59. Left_join_records_on()



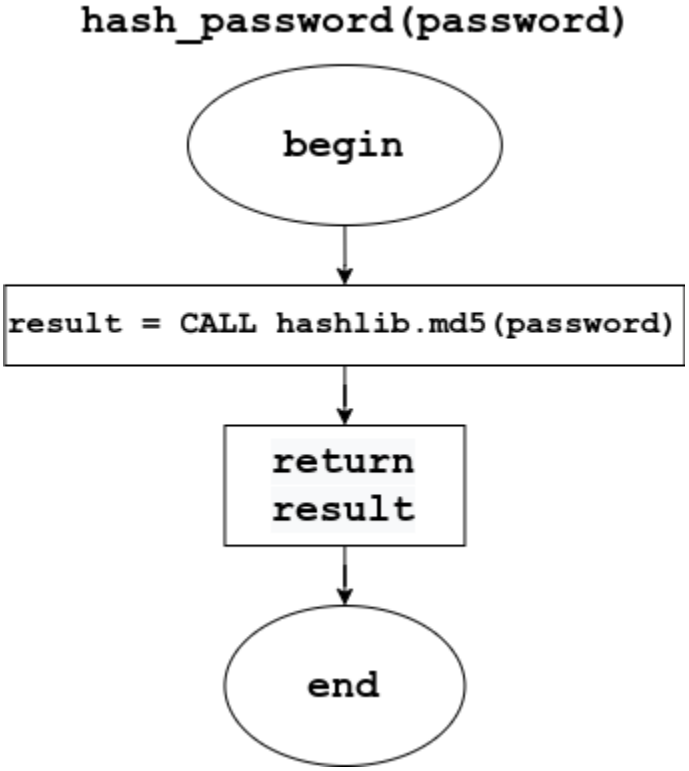
60. Hhmm_to_seconds()



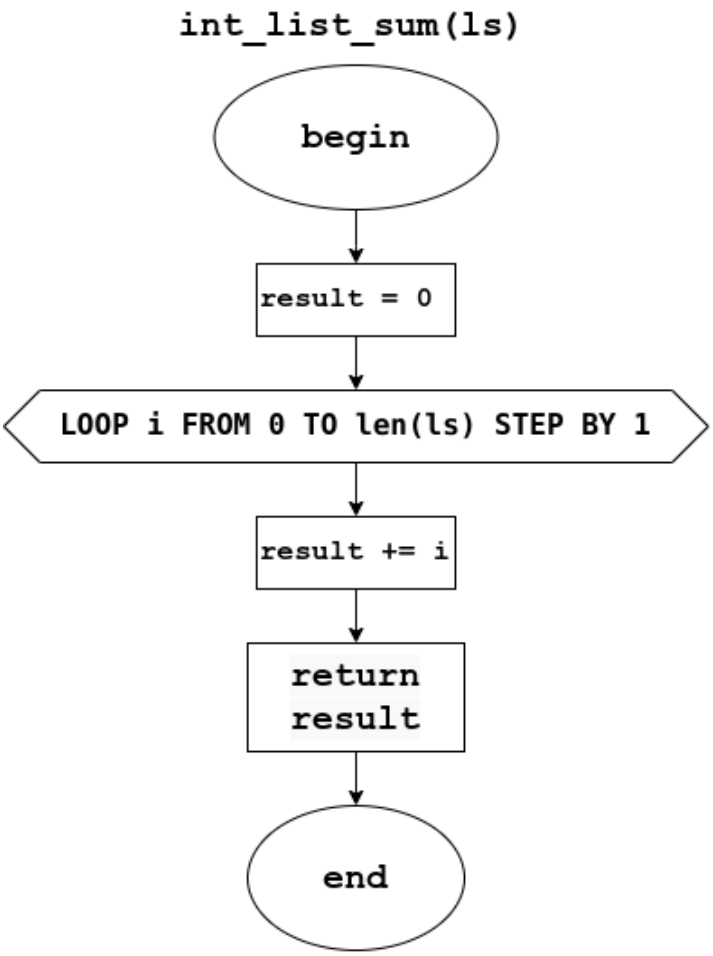
61. Get_date()



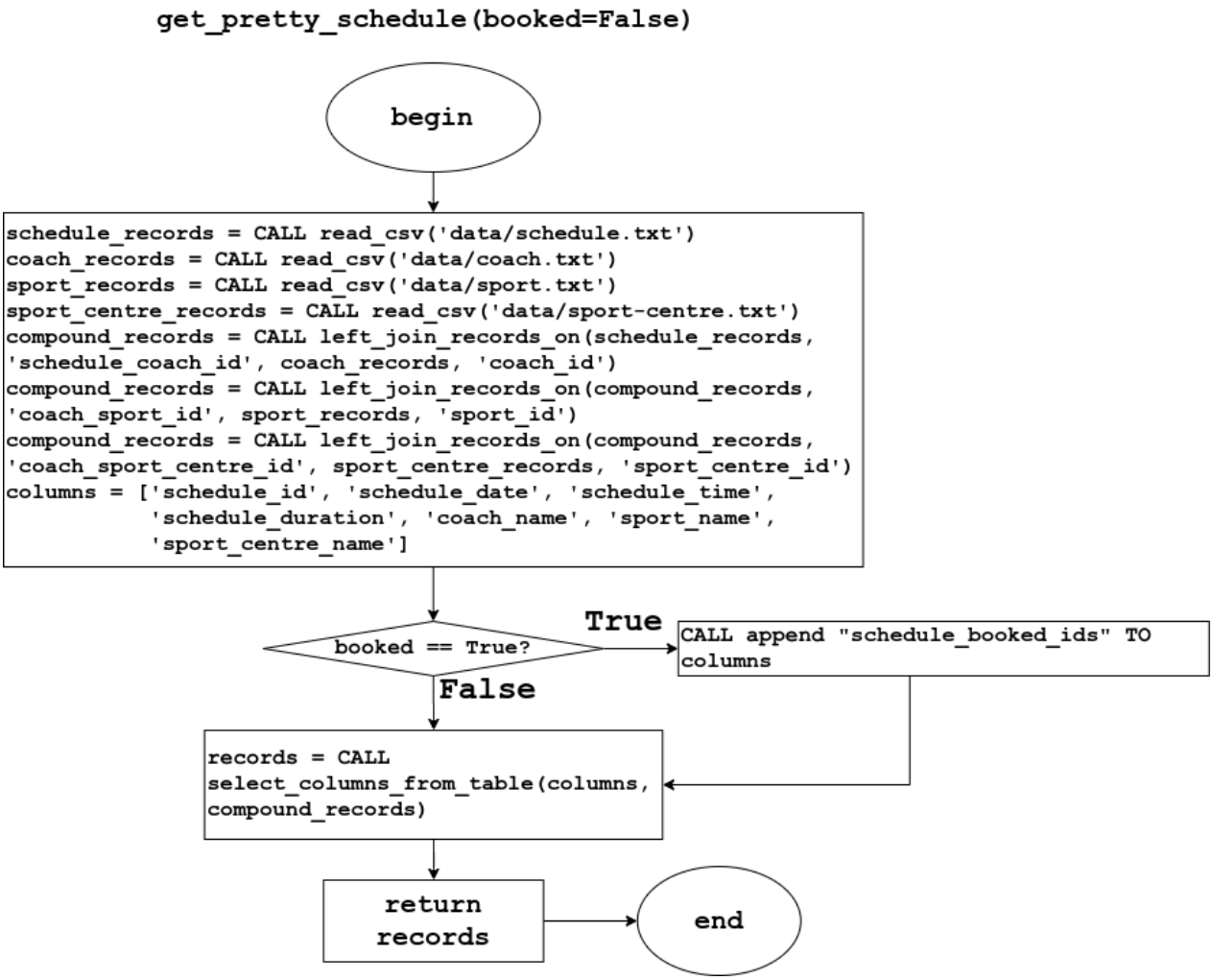
62. Hash_password()



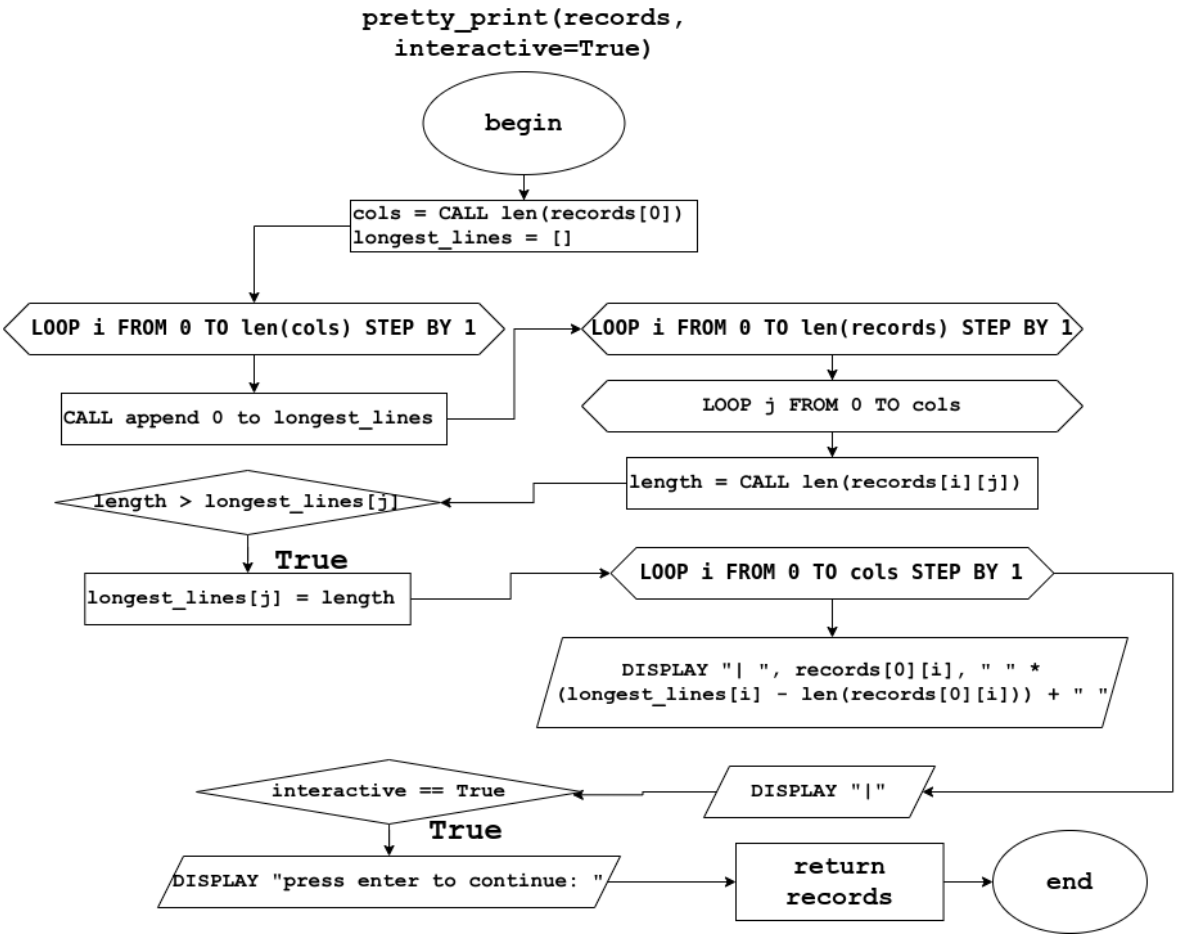
63. Int_list_sum()



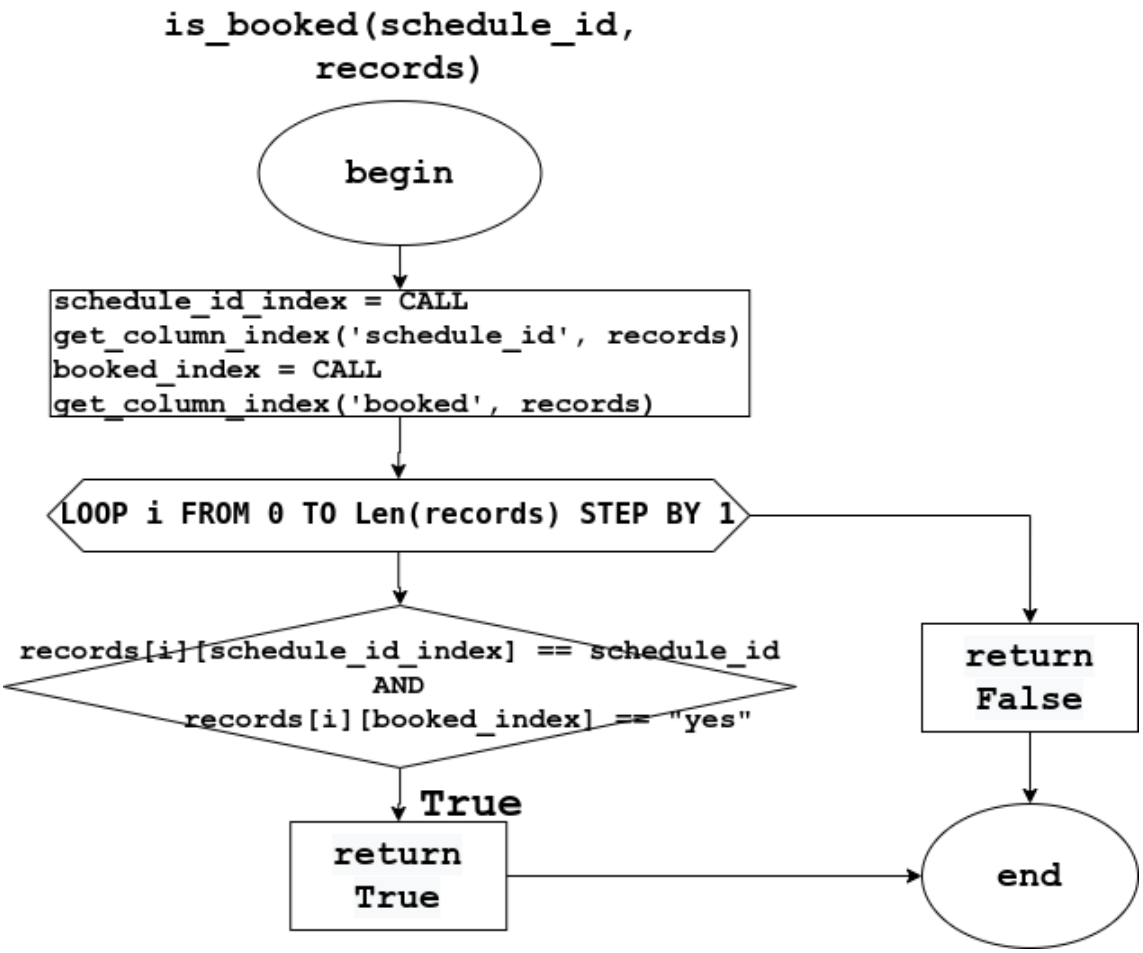
64. Get_pretty_schedule()



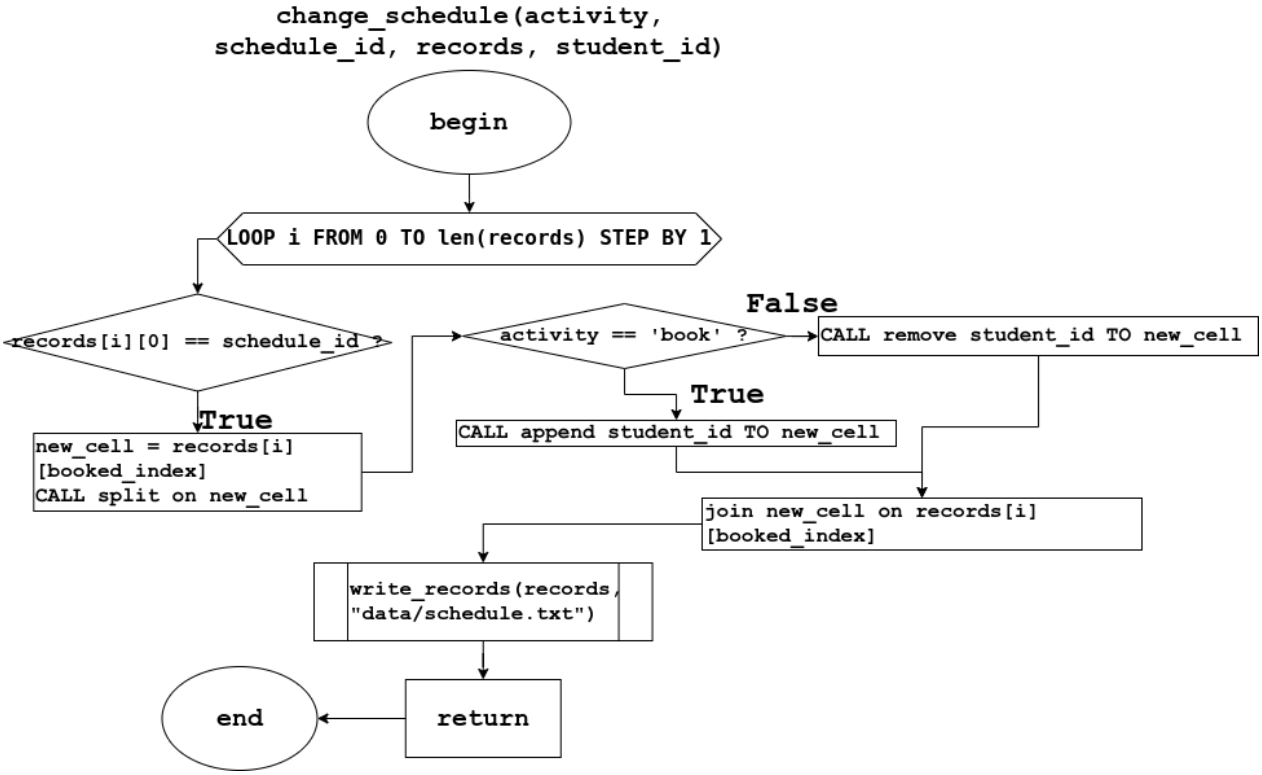
65. Pretty_print()



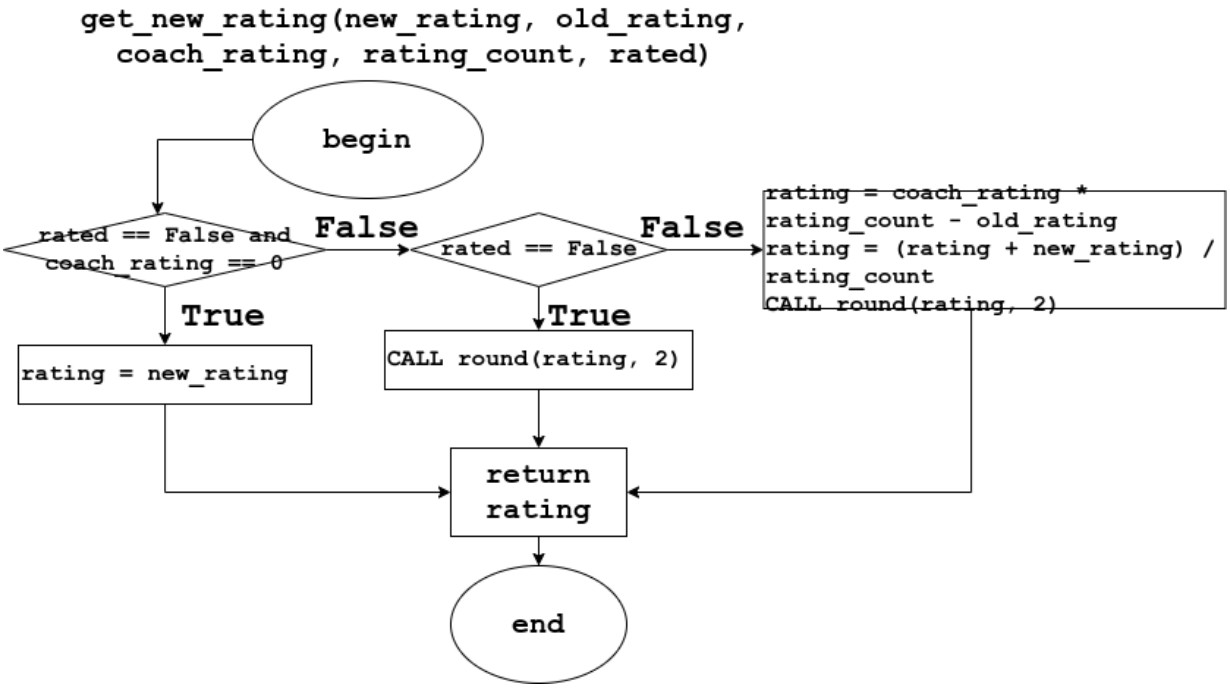
66. Is_booked()


is_booked(schedule_id, records)
begin
schedule_id_index = CALL get_column_index('schedule_id', records)
booked_index = CALL get_column_index('booked', records)
LOOP i FROM 0 TO Len(records) STEP BY 1
records[i][schedule_id_index] == schedule_id AND records[i][booked_index] == "yes"
True
return True
return False
end

67. Change_schedule()



68. Get_new_rating()



Sample Python Code

Variables

Variables are identifiers for memory locations to store data (TutorialsPoint, n.d.). When you assign a value to a variable, that data is stored somewhere in memory, and the variable name is used to retrieve the stored value. Variables in python can be in several distinct types. The types are integer, float (floating point), string (text), bool (Boolean values), and list. Variables are defined with the "=" operator. The syntax for variable definition is "<variable name> = <value>".

```
coach_sport_records = left_join_records_on(coach_records, 'coach_sport_id',
                                           sport_records, 'sport_id')
columns = ['coach_id', 'coach_name', 'coach_date_joined',
           'coach_date_terminated', 'coach_phone', 'coach_address',
           'coach_sport_id', 'coach_sport_centre_id', 'coach_rating',
           'sport_name']
coach_sport_records = select_columns_from_table(columns, coach_sport_records)
search_records_by('sport_name', 'coach sport name', coach_sport_records)
```

Examples of variable definitions

Operators

Python supports different types of operators, including arithmetic operators, comparison operators, assignment operators, logical operators, etc. (TutorialsPoint, n.d.). the arithmetic operators include addition "+", subtraction "-", multiplication "*", division "/", modulus operator "%", etc. Comparison operators include less than "<", greater than ">', less than equal, "<=", greater than equal ">=", equal "==", and not equal "!=". Logical operators are "and", "or", and "not".

```
    if seconds ≥ lower_bound and seconds < upper_bound:
        records.append(i)
```

Comparison and logical operators in use

Conditional statements

In python, you can execute certain parts of code based on some conditions. This is done using conditional statements. Conditional statements in python uses the "if", "elif", and "else" keywords. There are several ways of decision making in python. The first is if statements. These statements execute code only if a condition is true.

```
672     if interactive:
673         input('press enter to continue: ')
674
```

Example of an if statement

Another type of conditional statement is the if else statement. In this type, there are 2 code sections that can be run. If the condition is true, then the code under the “if” keyword will be executed. If the condition is false, the code under the “else” keyword is executed.

```
if activity == 'book':
    new_cell.append(student_id)
else:
    new_cell.remove(student_id)
```

Example of an if else statement

The other type of conditional statement is the nested if statement. In this type, conditional statements are nested inside another one. This allows for deciding between more than two possible choices.

```
# execute choice
if inp == '1':
    admin_add_records()
elif inp == '2':
    admin_display_records()
elif inp == '3':
    admin_search_records()
elif inp == '4':
    admin_sort_records()
elif inp == '5':
    admin_modify_records()
else:
    return
```

Example of a nested if else statement

Lists

The list is one of the basic data structures in python (Tutorialspoint, n.d.). Lists in python are comma-separated values in between square brackets. Lists are ordered and can be indexed or sliced using square brackets. Lists in python can be modified, which means their items can be replaced with another value.

```
columns = ['coach_id', 'coach_name', 'coach_date_joined',
           'coach_date_terminated', 'coach_phone', 'coach_address',
           'coach_sport_id', 'coach_sport_centre_id', 'coach_rating',
           'sport_name']
```

Screenshot of a list

```
ihour = int(inp[:2])
imin = int(inp[3:])
```

Example of list slicing

Looping statements

In python, we can execute the same code multiple times using loop structures. loops can be used until a condition is fulfilled, or for iterating over objects in a list. There are 2 types of loops we can use in python, while loops and for loops. While loops execute the same code until a certain condition is met. If the condition isn't met, the loop will run indefinitely, this is sometimes called an infinite loop.

```
# validate login
max_tries = 4
while validate_login(email, password) == False and max_tries > 0:
    print('invalid credentials, you have ' + str(max_tries) + ' tries left')
    email = input('enter your email: ')
    email = validate_empty_input(email, 'email')

    password = input('enter your password: ')
    password = validate_empty_input(password, password)

    max_tries -= 1
```

Example of while loop usage

The other type of loop is the for loop. It usually used for iterating over a list or a range of numbers. Unlike the while loop, no condition is needed, the loop will terminate once it has gone through all the items.

```
result = []
for i in records:
    result.append(i[index])
```

This for loop iterates through all items in the variable records and appends them to the variable result.

Just like conditional statements, loops can also be nested inside one another. They are referred to as nested loops.

```
n = len(records)
for i in range(n - 1):
    for j in range(n - i - 1):
        if records[j][index] > records[j + 1][index]:
            temp =  records[j + 1]
            records[j + 1] = records[j]
            records[j] = temp
```

Example of a nested loop structure

Functions

A function is a section of code that is reusable and can be called multiple times. Python provides many built-in functions such as print(), input(), len(), range(), etc. We can also define our own functions using the “def” keyword. A function can also have parameters and return values. A parameter is a positional or named input that is provided to the function, while the return value is the value of the function after it is being called.

```
258 def validate_login(email, password):
259     student_records = read_csv('data/student.txt')
260     student_records = select_columns_from_table(['student_email', 'student_password'], student_records)
261 
262     for i in student_records:
263         if i[0] == email and hash_password(password) == i[1]:
264             return True
265     return False
```

Example of a function with 2 parameters and a Boolean return type

```
index = get_column_index(col_name, records)
```

Example of a function call. the return value is assigned to the variable “index”.

File I/O

Python provides built-in functions to handle input and output. The print() function is used to print text to STDOUT (standard output), while the input() function is used to receive input from users from STDIN (standard input).

```
print('1. Display Coach Records\n2. Display Student Records')
print('3. Display Schedule Records\n4. Display Sport Records')
print('5. Display Sport Centre Records\n6. Return to Admin Menu')

# get and validate input
inp = input('enter choice: ')
```

Print() and input() usage

Python also has built-in functions for file operations. To read or write from/to files, we can use the open() and close() functions. The first thing to do to access a file through python is to

open it. We can specify the purpose of opening the file by specifying the open mode in the function call. The types of modes include “r” (read), “w” (write), “a” (append), etc. After we’re finished using the file, we need to close it using the close() function.

```
434 def write_record(record, filename, interactive=True):
435     record = ','.join(record)
436     file = open(filename, 'a'):
437         file.write(record + '\n')
438     if interactive:
439         input('Succesfully added record. Press enter to continue: ')
440     close(file)
441
```

Example of file operations

Modules

A module is a file containing python code that can be imported to other code (TutorialsPoint, n.d.). We can import modules by using the “import” keyword. Python comes with several modules by default such as the “random” module, “string” module, etc.

```
4 from datetime import datetime
5 import hashlib
6 import os
7
```

Example of importing modules

Comments

Comments are the part of a source code that is not executed by the interpreter or compiled by the compiler. They are useful for documenting code and explaining complicated sections. Comments in python start with a “#”. There are also docstrings in python, which are used specifically to document functions or classes. These start and end with triple quotes (‘’').

```
    for i in range(len(records)):
        for j in range(len(records[i])):
            # change '-' to empty string
            if records[i][j] == '-':
                records[i][j] = ''
        records[i] = ','.join(records[i]) + '\n'

    with open(filename, 'w') as file:
        file.writelines(records)
```

Example of comments

```
434 def write_record(record, filename, interactive=True):
435     '''append new record to filename
436 
437     parameters:
438     record (list): record (row) to append to records
439     filename (str): filename of records to append to
440     interactive (bool): set False if pause after write is not needed
441     '''
442 
443     record = ','.join(record)
444     with open(filename, 'a') as file:
445         file.write(record + '\n')
446     if interactive:
447         input('Succesfully added record. Press enter to continue: ')
448
```

Example of docstring usage

Usage Screenshot

1. Main menu, shown after running program.

```
REAL CHAMPIONS SPORT CENTRE
1. Admin
2. Student
3. Quit
enter choice:
```

1. Admin login function, shown after entering 1 on main menu

```
ADMIN LOGIN
enter password:
```

2. Wrong input in admin login function

```
ADMIN LOGIN
enter password: j
wrong password, you have 4 tries left
enter password: kjs
wrong password, you have 3 tries left
enter password: ks
wrong password, you have 2 tries left
enter password: k
wrong password, you have 1 tries left
enter password:
```

3. Admin menu function

```
ADMIN MENU
1. Add Records
2. Display Records
3. Search Specific Record
4. Sort Records
5. Modify Records
6. Quit
enter choice:
```

4. Admin add records function menu, shown after entering 1 on admin menu

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice:
```

5. Admin add coach records sample input, after enterin 1 on add records menu

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice: 1
enter coach name: cooper
enter date joined (in DD-MM-YYYY format), leave blank for current date:
enter date terminated (leave blank to skip):
enter coach phone number: 092849
enter coach address: kuala lumpur
enter sport name: blablabla
invalid sport, enter a valid sport: badminton
enter sport centre name: sport centre A
Succesfully added record. Press enter to continue:
```

6. Admin add schedule records given invalid data, after entering 2 on add records menu

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice: 2
enter class date (in DD-MM-YYYY), leave blank for current date:
enter class time (24-hour format HH:MM): 09:00
enter duration (between 1-9 hours): 10
invalid duration, duration must be between 1-9 and class must not pass 17:00
enter valid duration: 2
enter coach id: 1
coach is already booked! press enter to continue:
```

7. Admin add schedule records given valid data

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice: 2
enter class date (in DD-MM-YYYY), leave blank for current date: 23-01-2021
enter class time (24-hour format HH:MM): 08:00
enter duration (between 1-9 hours): 3
enter coach id: 7
Succesfully added record. Press enter to continue:
```

8. Admin add sport records

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice: 3
enter sport name: chess
enter hourly fee: 400
Succesfully added record. Press enter to continue:
```

9. Admin add sport centre records

```
ADD RECORDS
1. Add Coach Record
2. Add Schedule Record
3. Add Sport Record
4. Add Sport Centre Record
5. Return to Admin Menu
enter choice: 4
enter sport centre name: sport centre d
enter location: kuala lumpur
Succesfully added record. Press enter to continue:
```

10. Admin display records function menu

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice:
```

11. Admin display coach records

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice: 1
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John        | 12-01-2020        | -                     | 1929384      | 22 baker street | 1              | 1                     | 4.25         |
| 2        | watson      | 12-12-2020        | -                     | 1231         | nowhere         | 2              | 2                     | 5.0          |
| 3        | martin      | 12-01-2020        | -                     | 9384         | nn              | 3              | 3                     | 5.0          |
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain         | 6              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123             | 7              | 3                     | 0            |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf    | 7              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            |
press enter to continue:
```

12. Admin display student records

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice: 2
| student_id | student_name | student_date_joined | student_email        | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------
| 1          | martin       | 12-12-2020          | 1@email.com          | e10adc3949ba59abbe56e057f20f883e | 09834         | nowher          |
| 2          | kk           | 12-12-2020          | kk@email.com         | 25d55ad283aa400af464c76d713c07ad | 123           | -               |
| 3          | john         | 12-12-2020          | johnsmith@email.com  | 7b8b965ad4bca0e41ab51de7b31363a1 | 123           | 123             |
| 4          | stark        | 12-12-2020          | tonystark@email.com  | 25d55ad283aa400af464c76d713c07ad | -             | -               |
| 5          | jjj          | 12-12-2020          | a                    | 7b8b965ad4bca0e41ab51de7b31363a1 | -             | -               |
| 6          | ryan         | 20-01-2021          | ryan.mrt1n@gmail.com | 4e3ab19467222ea9ab20c92cb311eb35 | -             | -               |
press enter to continue:
```

13. Admin display schedule records

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice: 3
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-----------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | 1                 | 3:4                 |
| 2           | 12-12-2020    | 13:00         | 1                 | 2                 | -                   |
| 3           | 12-12-2020    | 15:00         | 2                 | 3                 | -                   |
| 4           | 18-01-2021    | 09:00         | 8                 | 7                 | -                   |
| 5           | 22-01-2021    | 09:00         | 3                 | 1                 | -                   |
| 6           | 23-01-2021    | 08:00         | 3                 | 7                 | -                   |
press enter to continue:
```

14. Admin display sports records

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice: 4
| sport_id | sport_name   | sport_hourly_fee |
-----------------------------------------------
| 1        | badminton    | 200              |
| 2        | swimming     | 300              |
| 3        | football     | 400              |
| 4        | archery      | 300              |
| 5        | tennis       | 300              |
| 6        | table tennis | 100              |
| 7        | volleyball   | 200              |
| 8        | basketball   | 300              |
| 9        | cricket      | 500              |
| 10       | gymnastics   | 200              |
| 11       | shogi        | 500              |
| 12       | chess        | 400              |
press enter to continue:
```

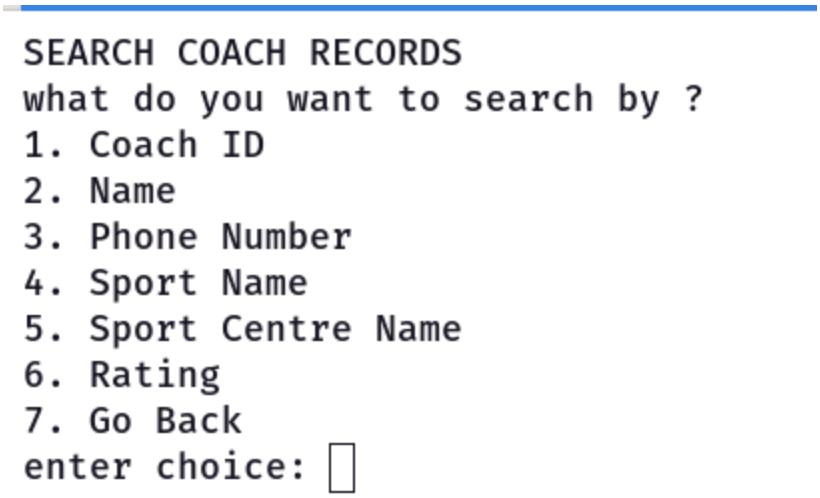
15. Admin display sport centre records

```
DISPLAY RECORDS
1. Display Coach Records
2. Display Student Records
3. Display Schedule Records
4. Display Sport Records
5. Display Sport Centre Records
6. Return to Admin Menu
enter choice: 5
| sport_centre_id | sport_centre_name | sport_centre_location |
-------------------------------------------------------------
| 1               | sport centre a    | kuala lumpur          |
| 2               | sport centre b    | johor bahru           |
| 3               | sport centre c    | penang                |
| 4               | sport centre d    | melaka                |
| 5               | sport centre d    | kuala lumpur          |
press enter to continue:
```

16. Admin search records function menu

```
SEARCH RECORDS
1. Search for Coach Record
2. Search for Student Record
3. Search for Schedule Record
4. Search for Sport Record
5. Search for Sport Centre Record
6. Return to Admin Menu
enter choice:
```

17. Admin search coach records menu



18. Admin search coach by id

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 1
enter coach id: 7
| coach_id | coach_name | coach_date_joined | coach_date_terminated | coach_phone | coach_address | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------------
| 7        | mage       | 18-01-2021        | -                     | 21398       | fdlskjfalkjf  | 7              | 2                     | 0            |
press enter to go back:
```

19. Admin search coach by name

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 2
enter coach name: cooper
| coach_id | coach_name | coach_date_joined | coach_date_terminated | coach_phone | coach_address | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------------
| 8        | cooper     | 22-01-2021        | -                     | 092849      | kuala lumpur  | 1              | 1                     | 0            |
press enter to go back:
```

20. Admin search coach by phone number

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 3
enter coach phone number: 123
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone | coach_address | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------------
| 4        | cumberbatch | 12-12-2019        | -                     | 123         | britain       | 6              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123         | 123           | 7              | 3                     | 0            |
press enter to go back:
```

21. Admin search coach by sports they teach

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 4
enter coach sport name: badminton
| coach_id | coach_name | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating | sport_name |
-------------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John       | 12-01-2020        | -                     | 1929384      | 22 baker street | 1              | 1                     | 4.25         | badminton  |
| 6        | tony       | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            | badminton  |
| 8        | cooper     | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            | badminton  |
press enter to go back:
```

22. Admin search coach by sport centre

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 5
enter coach sport centre: sport centre a
| coach_id | coach_name | coach_date_joined | coach_date_terminated | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating | sport_centre_name |
-------------------------------------------------------------------------------------------------------------------------------------
| 1        | John       | 12-01-2020        | -                     | 22 baker street | 1              | 1                     | 4.25         | sport centre a    |
| 8        | cooper     | 22-01-2021        | -                     | kuala lumpur    | 1              | 1                     | 0            | sport centre a    |
press enter to go back:
```

23. Admin search coach by rating

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 6
enter coach rating: 0
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain       | 6              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123           | 7              | 3                     | 0            |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia      | 1              | 3                     | 0            |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf  | 7              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur  | 1              | 1                     | 0            |
press enter to go back:
```

24. Admin search function given invalid value

```
SEARCH COACH RECORDS
what do you want to search by ?
1. Coach ID
2. Name
3. Phone Number
4. Sport Name
5. Sport Centre Name
6. Rating
7. Go Back
enter choice: 1
enter coach id: 10
there is no record with coach id of 10
press enter to go back:
```

25. Admin search student records menu

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice:
```

26. Admin search student by student id

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice: 1
enter student id: 1
| student_id | student_name | student_date_joined | student_email | student_password                | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------------
| 1          | martin       | 12-12-2020          | 1@email.com   | e10adc3949ba59abbe56e057f20f883e | 09834         | nowher          |
press enter to go back:
```

27. Admin search student by name

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice: 2
enter student name: john
| student_id | student_name | student_date_joined | student_email       | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------
| 3          | john         | 12-12-2020          | johnsmith@email.com | 7b8b965ad4bca0e41ab51de7b31363a1 | 123           | 123             |
press enter to go back:
```

28. Admin search student by phone number

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice: 3
enter phone number: 09834
| student_id | student_name | student_date_joined | student_email | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------
| 1          | martin       | 12-12-2020          | 1@email.com   | e10adc3949ba59abbe56e057f20f883e | 09834         | nowher          |
press enter to go back:
```

29. Admin search student by email

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice: 4
enter email: johnsmith@email.com
| student_id | student_name | student_date_joined | student_email       | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------
| 3          | john         | 12-12-2020          | johnsmith@email.com | 7b8b965ad4bca0e41ab51de7b31363a1 | 123           | 123             |
press enter to go back:
```

30. Admin search student given invalid input

```
SEARCH STUDENT RECORDS
what do you want to search by ?
1. Student ID
2. Name
3. Phone Number
4. Email
5. Go Back
enter choice: 2
enter student name: oz
there is no record with student name of oz
press enter to go back:
```

31. Admin search schedule records menu

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice:
```

32. Admin search schedule by date

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice: 1
enter schedule date:
schedule date must not be empty, enter schedule date: 22-01-2021
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-----------------------------------------------------------------------------------------------------------
| 5           | 22-01-2021    | 09:00         | 3                 | 1                 | -                   |
press enter to go back:
```

33. Admin search schedule by time

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice: 2
enter class time (HH:MM 24-hour format): 09:37
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-----------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | 1                 | 3:4                 |
| 4           | 18-01-2021    | 09:00         | 8                 | 7                 | -                   |
| 5           | 22-01-2021    | 09:00         | 3                 | 1                 | -                   |
| 6           | 23-01-2021    | 08:00         | 3                 | 7                 | -                   |
press enter to go back:
```

34. Admin search schedule by duration

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice: 3
enter schedule duration: 8
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-------------------------------------------------------------------------------------------------------------------
| 4           | 18-01-2021    | 09:00         | 8                 | 7                 | -                   |
press enter to go back:
```

35. Admin search schedule by coach id

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice: 4
enter coach id: 1
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-------------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | 1                 | 3:4                 |
| 5           | 22-01-2021    | 09:00         | 3                 | 1                 | -                   |
press enter to go back:
```

36. Admin search schedule given invalid input

```
SEARCH SCHEDULE RECORDS
what do you want to search by ?
1. Date
2. Time
3. Duration
4. Coach ID
5. Go Back
enter choice: 1
enter schedule date: 01-01-2001
there is no record with schedule date of 01-01-2001
press enter to go back:
```

37. Admin search sport records function menu

```
what do you want to search by ?
1. Sport ID
2. Name
3. Hourly fee
4. Go Back
enter choice:
```

38. Admin search sport by sport id

```
what do you want to search by ?
1. Sport ID
2. Name
3. Hourly fee
4. Go Back
enter choice: 1
enter sport id: 8
| sport_id | sport_name | sport_hourly_fee |
-------------------------------------------
| 8        | basketball | 300              |
press enter to continue:
```

39. Admin search sport by sport name

```
what do you want to search by ?
1. Sport ID
2. Name
3. Hourly fee
4. Go Back
enter choice: 2
enter sport name: tennis
| sport_id | sport_name | sport_hourly_fee |
-------------------------------------------
| 5        | tennis     | 300              |
press enter to continue:
```

40. admin search sport by hourly fee

```
what do you want to search by ?
1. Sport ID
2. Name
3. Hourly fee
4. Go Back
enter choice: 3
enter hourly fee: 200
| sport_id | sport_name | sport_hourly_fee |
-------------------------------------------
| 1        | badminton  | 200              |
| 7        | volleyball | 200              |
| 10       | gymnastics | 200              |
press enter to continue:
```

41. Admin search sport given invalid input

```
what do you want to search by ?
1. Sport ID
2. Name
3. Hourly fee
4. Go Back
enter choice: 1
enter sport id: 19
there is no record with sport id of 19
press enter to continue:
```

42. Admin search sport centre records menu

```
SEARCH SPORT CENTRE RECORDS
what do you want to search by ?
1. Sport Centre ID
2. Name
3. Location
4. Go Back
enter choice:
```

43. Admin search sport centre by sport centre id

```
SEARCH SPORT CENTRE RECORDS
what do you want to search by ?
1. Sport Centre ID
2. Name
3. Location
4. Go Back
enter choice: 1
enter sport centre id: 2
| sport_centre_id | sport_centre_name | sport_centre_location |
-----------------------------------------------------------------
| 2               | sport centre b    | johor bahru           |
press enter to continue:
```

44. Admin search sport centre by name

```
SEARCH SPORT CENTRE RECORDS
what do you want to search by ?
1. Sport Centre ID
2. Name
3. Location
4. Go Back
enter choice: 2
enter sport centre name: Sport Centre A
| sport_centre_id | sport_centre_name | sport_centre_location |
-----------------------------------------------------------------
| 1               | sport centre a    | kuala lumpur          |
press enter to continue:
```

45. Admin search sport centre by location

```
SEARCH SPORT CENTRE RECORDS
what do you want to search by ?
1. Sport Centre ID
2. Name
3. Location
4. Go Back
enter choice: 2
enter sport centre name: Sport Centre A
| sport_centre_id | sport_centre_name | sport_centre_location |
-----------------------------------------------------------------
| 1               | sport centre a    | kuala lumpur          |
press enter to continue:
```

46. Admin search sport centre given invalid input

```
SEARCH SPORT CENTRE RECORDS
what do you want to search by ?
1. Sport Centre ID
2. Name
3. Location
4. Go Back
enter choice: 3
enter location:
location must not be empty, enter location:
location must not be empty, enter location: s
there is no record with location of s
press enter to continue:
```

47. Admin sort records function menu

```
DISPLAY SORTED RECORDS
1. Sort Coach by Coach ID
2. Sort Coach by Name
3. Sort Coach by Hourly Fee
4. Sort Sport by Hourly Fee
5. Return to Admin Menu
enter choice:
```

48. Admin sort coach by id

```
DISPLAY SORTED RECORDS
1. Sort Coach by Coach ID
2. Sort Coach by Name
3. Sort Coach by Hourly Fee
4. Sort Sport by Hourly Fee
5. Return to Admin Menu
enter choice: 1
display records in ascending order? [Y/n]: y
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John        | 12-01-2020        | -                     | 1929384      | 22 baker street | 1              | 1                     | 4.25         |
| 2        | watson      | 12-12-2020        | -                     | 1231         | nowhere         | 2              | 2                     | 5.0          |
| 3        | martin      | 12-01-2020        | -                     | 9384         | nn              | 3              | 3                     | 5.0          |
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain         | 6              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123             | 7              | 3                     | 0            |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf    | 7              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            |
press enter to continue:
```

49. Admin sort coach by name

```
DISPLAY SORTED RECORDS
1. Sort Coach by Coach ID
2. Sort Coach by Name
3. Sort Coach by Hourly Fee
4. Sort Sport by Hourly Fee
5. Return to Admin Menu
enter choice: 2
display records in ascending order? [Y/n]: n
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------------
| 2        | watson      | 12-12-2020        | -                     | 1231         | nowhere         | 2              | 2                     | 5.0          |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            |
| 3        | martin      | 12-01-2020        | -                     | 9384         | nn              | 3              | 3                     | 5.0          |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf    | 7              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123             | 7              | 3                     | 0            |
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain         | 6              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            |
| 1        | John        | 12-01-2020        | -                     | 1929384      | 22 baker street | 1              | 1                     | 4.25         |
press enter to continue:
```

50. Admin sort coach by hourly fee

```
DISPLAY SORTED RECORDS
1. Sort Coach by Coach ID
2. Sort Coach by Name
3. Sort Coach by Hourly Fee
4. Sort Sport by Hourly Fee
5. Return to Admin Menu
enter choice: 3
display records in ascending order? [Y/n]: y
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | sport_name   | coach_sport_centre_id | coach_rating | sport_hourly_fee |
-----------------------------------------------------------------------------------------------------------------------------------------
| 4        | cumberbatch | 12-12-2019        | -                     | table tennis | 2                     | 0            | 100              |
| 1        | John        | 12-01-2020        | -                     | badminton    | 1                     | 4.25         | 200              |
| 5        | jo          | 11-11-2010        | -                     | volleyball   | 3                     | 0            | 200              |
| 6        | tony        | 01-01-2021        | -                     | badminton    | 3                     | 0            | 200              |
| 7        | mage        | 18-01-2021        | -                     | volleyball   | 2                     | 0            | 200              |
| 8        | cooper      | 22-01-2021        | -                     | badminton    | 1                     | 0            | 200              |
| 2        | watson      | 12-12-2020        | -                     | swimming     | 2                     | 5.0          | 300              |
| 3        | martin      | 12-01-2020        | -                     | football     | 3                     | 5.0          | 400              |
press enter to continue:
```

51. Admin sort sport by hourly fee

```
DISPLAY SORTED RECORDS
1. Sort Coach by Coach ID
2. Sort Coach by Name
3. Sort Coach by Hourly Fee
4. Sort Sport by Hourly Fee
5. Return to Admin Menu
enter choice: 4
display records in ascending order? [Y/n]: n
| sport_id | sport_name   | sport_hourly_fee |
-------------------------------------------
| 11       | shogi        | 500              |
| 9        | cricket      | 500              |
| 12       | chess        | 400              |
| 3        | football     | 400              |
| 8        | basketball   | 300              |
| 5        | tennis       | 300              |
| 4        | archery      | 300              |
| 2        | swimming     | 300              |
| 10       | gymnastics   | 200              |
| 7        | volleyball   | 200              |
| 1        | badminton    | 200              |
| 6        | table tennis | 100              |
press enter to continue:
```

52. Admin modify records function menu

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice:
```

53. Admin modify coach records

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice: 1
enter coach id to modify: 1
enter new coach name (leave blank to skip):
enter new date joined (leave blank to skip):
enter new date terminated (leave blank to skip): 22-01-2021
enter new coach phone number (leave blank to skip):
enter new coach address (leave blank to skip):
enter new sport id (leave blank to skip):
enter new sport centre id (leave blank to skip):
enter new coach rating (leave blank to skip):
| coach_id | coach_name | coach_date_joined | coach_date_terminated | coach_phone | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-----------------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John       | 12-01-2020        | 22-01-2021            | 1929384     | 22 baker street | 1              | 1                     | 4.25         |
enter y to change, n to cancel: y
Succesfully modified record. Press enter to continue:
```

54. Admin modify schedule records invalid input

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice: 2
enter schedule id to modify: 1
enter new class date (leave blank to skip):
enter new class time (leave blank to skip):
enter new duration (between 1-9 hours): 9
enter new coach id (leave blank to skip): 2
invalid schedule. press enter to try again:
```

55. Admin modify schedule records given valid input

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice: 2
enter schedule id to modify: 1
enter new class date (leave blank to skip):
enter new class time (leave blank to skip):
enter new duration (between 1-9 hours): 9
enter new coach id (leave blank to skip): 1
| schedule_id | schedule_date | schedule_time | schedule_duration | schedule_coach_id | schedule_booked_ids |
-------------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 9                 | 1                 | 3:4                 |
enter y to change, n to cancel:
```

56. Admin modify sport records

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice: 3
enter sport id to modify: 1
enter new sport name (leave blank to skip):
enter new hourly fee (leave blank to skip): 100
| sport_id | sport_name | sport_hourly_fee |
-------------------------------------------
| 1        | badminton  | 100              |
enter y to change, n to cancel:
```

57. Admin modify sport centre records

```
MODIFY RECORDS
1. Modify Coach Records
2. Modify Schedule Records
3. Modify Sport Records
4. Modify Sport Centre Records
5. Return to Admin Menu
enter choice: 4
enter sport centre id to modify: 1
enter new sport centre name (leave blank to skip):
enter new location (leave blank to skip): melaka
| sport_centre_id | sport_centre_name | sport_centre_location |
-----------------------------------------------------------------
| 1               | sport centre a    | melaka                |
enter y to change, n to cancel:
```

58. Unregistered student menu

```
STUDENT MENU
1. View Details
2. Login
3. Register
4. Quit
enter choice:
```

59. Unregistered student view details menu

```
VIEW DETAILS
1. View Sport Details
2. View Sport Schedule
3. Return to Student Menu
enter choice:
```

60. Unregistered student view sport details

```
VIEW DETAILS
1. View Sport Details
2. View Sport Schedule
3. Return to Student Menu
enter choice: 1
| sport_id | sport_name   | sport_hourly_fee |
-------------------------------------------
| 1        | badminton    | 200              |
| 2        | swimming     | 300              |
| 3        | football     | 400              |
| 4        | archery      | 300              |
| 5        | tennis       | 300              |
| 6        | table tennis | 100              |
| 7        | volleyball   | 200              |
| 8        | basketball   | 300              |
| 9        | cricket      | 500              |
| 10       | gymnastics   | 200              |
| 11       | shogi        | 500              |
| 12       | chess        | 400              |
press enter to continue:
```

61. Unregistered student view sport schedule

```
VIEW DETAILS
1. View Sport Details
2. View Sport Schedule
3. Return to Student Menu
enter choice: 2
| schedule_id | schedule_date | schedule_time | schedule_duration | coach_name | sport_name | sport_centre_name |
-----------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | John       | badminton  | sport centre a    |
| 2           | 12-12-2020    | 13:00         | 1                 | watson     | swimming   | sport centre b    |
| 3           | 12-12-2020    | 15:00         | 2                 | martin     | football   | sport centre c    |
| 4           | 18-01-2021    | 09:00         | 8                 | mage       | volleyball | sport centre b    |
| 5           | 22-01-2021    | 09:00         | 3                 | John       | badminton  | sport centre a    |
| 6           | 23-01-2021    | 08:00         | 3                 | mage       | volleyball | sport centre b    |
press enter to continue:
```

62. Student login with invalid credentials

```
STUDENT LOGIN
enter your email: johnsmith@email.com
enter your password:
 must not be empty, enter : a
invalid credentials, you have 4 tries left
enter your email: johnsmith@email.com
enter your password: sdf
invalid credentials, you have 3 tries left
enter your email: dfa
enter your password: d
invalid credentials, you have 2 tries left
enter your email:
```

63. Unregistered student register

```
STUDENT REGISTER
enter your name: sock
enter your email address: sock@email.com
enter your password (minimum 8 characters): jjj
password must be atleast 8 characters!
enter password: 12345678
enter your phone number (leave blank to skip):
enter your address (leave blank to skip):
Succesfully added record. Press enter to continue:
```

64. Registered student menu

```
STUDENT MENU
1. View Details
2. Modify Self Record
3. Provide Coach Feedback
4. Manage Schedule
5. Quit
enter choice:
```

65. Registered student view details

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice:
```

66. Registered student view sport details

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 1
| sport_id | sport_name   | sport_hourly_fee |
-----------------------------------------------
| 1        | badminton    | 200              |
| 2        | swimming     | 300              |
| 3        | football     | 400              |
| 4        | archery      | 300              |
| 5        | tennis       | 300              |
| 6        | table tennis | 100              |
| 7        | volleyball   | 200              |
| 8        | basketball   | 300              |
| 9        | cricket      | 500              |
| 10       | gymnastics   | 200              |
| 11       | shogi        | 500              |
| 12       | chess        | 400              |
press enter to continue:
```

67. Registered student view profile

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 2
| student_id | student_name | student_date_joined | student_email       | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------------
| 3          | john         | 12-12-2020          | johnsmith@email.com | 7b8b965ad4bca0e41ab51de7b31363a1 | 123           | 123             |
press enter to continue:
```

68. Registered student view coach details

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 3
| coach_id | coach_name  | coach_phone  | sport_name    | sport_centre_name | coach_rating |
-----------------------------------------------------------------------------------------
| 1        | John        | 1929384      | badminton     | sport centre a    | 4.25         |
| 2        | watson      | 1231         | swimming      | sport centre b    | 5.0          |
| 3        | martin      | 9384         | football      | sport centre c    | 5.0          |
| 4        | cumberbatch | 123          | table tennis  | sport centre b    | 0            |
| 5        | jo          | 123          | volleyball    | sport centre c    | 0            |
| 6        | tony        | 082383749283 | badminton     | sport centre c    | 0            |
| 7        | mage        | 21398        | volleyball    | sport centre b    | 0            |
| 8        | cooper      | 092849       | badminton     | sport centre a    | 0            |
press enter to continue:
```

69. Registered student view sport schedule

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 4
| schedule_id | schedule_date | schedule_time | schedule_duration | coach_name | sport_name | sport_centre_name |
-----------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | John       | badminton  | sport centre a    |
| 2           | 12-12-2020    | 13:00         | 1                 | watson     | swimming   | sport centre b    |
| 3           | 12-12-2020    | 15:00         | 2                 | martin     | football   | sport centre c    |
| 4           | 18-01-2021    | 09:00         | 8                 | mage       | volleyball | sport centre b    |
| 5           | 22-01-2021    | 09:00         | 3                 | John       | badminton  | sport centre a    |
| 6           | 23-01-2021    | 08:00         | 3                 | mage       | volleyball | sport centre b    |
press enter to continue:
```

70. Registered student view booked sport schedule

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 5
| schedule_id | schedule_date | schedule_time | schedule_duration | coach_name | sport_name | sport_centre_name |
-----------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | John       | badminton  | sport centre a    |
press enter to continue:
```

71. Registered student modify self record

```
STUDENT MENU
1. View Details
2. Modify Self Record
3. Provide Coach Feedback
4. Manage Schedule
5. Quit
enter choice: 2
enter new name (leave blank to skip): johny
enter new email (leave blank to skip):
enter new phone (leave blank to skip):
enter new address (leave blank to skip):
enter new password (leave blank to skip):
| student_id | student_name | student_date_joined | student_email       | student_password                 | student_phone | student_address |
-------------------------------------------------------------------------------------------------------------------------------------------------
| 3          | johny        | 12-12-2020          | johnsmith@email.com | 7b8b965ad4bca0e41ab51de7b31363a1 | 123           | 123             |
enter y to change, n to cancel:
```

72. Registered student provide coach feedback menu

```
RATE COACH AND GIVE FEEDBACK
(feedbacks are anonymous)
1. Rate Coach
2. Provide Feedback
3. Return to Student Menu
enter choice:
```

73. Registered student rate coach

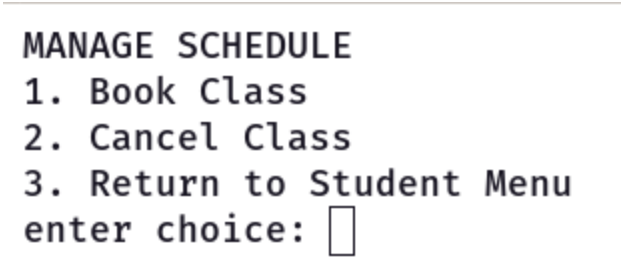
```
RATE COACH AND GIVE FEEDBACK
(feedbacks are anonymous)
1. Rate Coach
2. Provide Feedback
3. Return to Student Menu
enter choice: 1
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John        | 12-01-2020        | 22-01-2021            | 1929384      | 22 baker street | 1              | 1                     | 4.25         |
| 2        | watson      | 12-12-2020        | -                     | 1231         | nowhere         | 2              | 2                     | 5.0          |
| 3        | martin      | 12-01-2020        | -                     | 9384         | nn              | 3              | 3                     | 5.0          |
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain         | 6              | 2                     | 0            |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123             | 7              | 3                     | 0            |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf    | 7              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            |
enter coach id: 4
enter coach rating (from 1-5 points): 3
Succesfully modified record. Press enter to continue:
```

```
VIEW DETAILS
1. View Sport Details
2. View Profile
3. View Coach Details
4. View Sport Schedule
5. View Booked Classes
6. Return to Student Menu
enter choice: 3
| coach_id | coach_name  | coach_phone  | sport_name   | sport_centre_name | coach_rating |
-------------------------------------------------------------------------------------------
| 1        | John        | 1929384      | badminton    | sport centre a    | 4.25         |
| 2        | watson      | 1231         | swimming     | sport centre b    | 5.0          |
| 3        | martin      | 9384         | football     | sport centre c    | 5.0          |
| 4        | cumberbatch | 123          | table tennis | sport centre b    | 3.0          |
| 5        | jo          | 123          | volleyball   | sport centre c    | 0            |
| 6        | tony        | 082383749283 | badminton    | sport centre c    | 0            |
| 7        | mage        | 21398        | volleyball   | sport centre b    | 0            |
| 8        | cooper      | 092849       | badminton    | sport centre a    | 0            |
press enter to continue:
```

68. Registered student provide feedback

```
RATE COACH AND GIVE FEEDBACK
(feedbacks are anonymous)
1. Rate Coach
2. Provide Feedback
3. Return to Student Menu
enter choice: 2
| coach_id | coach_name  | coach_date_joined | coach_date_terminated | coach_phone  | coach_address   | coach_sport_id | coach_sport_centre_id | coach_rating |
-------------------------------------------------------------------------------------------------------------------------------------------
| 1        | John        | 12-01-2020        | 22-01-2021            | 1929384      | 22 baker street | 1              | 1                     | 4.25         |
| 2        | watson      | 12-12-2020        | -                     | 1231         | nowhere         | 2              | 2                     | 5.0          |
| 3        | martin      | 12-01-2020        | -                     | 9384         | nn              | 3              | 3                     | 5.0          |
| 4        | cumberbatch | 12-12-2019        | -                     | 123          | britain         | 6              | 2                     | 3.0          |
| 5        | jo          | 11-11-2010        | -                     | 123          | 123             | 7              | 3                     | 0            |
| 6        | tony        | 01-01-2021        | -                     | 082383749283 | malaysia        | 1              | 3                     | 0            |
| 7        | mage        | 18-01-2021        | -                     | 21398        | fdlskjfalkjf    | 7              | 2                     | 0            |
| 8        | cooper      | 22-01-2021        | -                     | 092849       | kuala lumpur    | 1              | 1                     | 0            |
enter coach id: 5
enter your feedback: you are a great teacher :v
Succesfully added record. Press enter to continue:
```

74. Registered student manage schedule



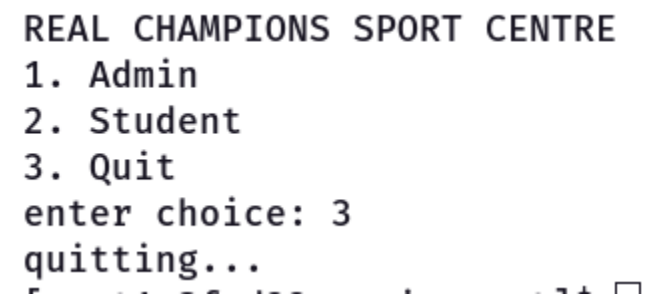
75. Registered student book class

```
MANAGE SCHEDULE
1. Book Class
2. Cancel Class
3. Return to Student Menu
enter choice: 1
| schedule_id | schedule_date | schedule_time | schedule_duration | coach_name | sport_name | sport_centre_name | booked |
-------------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | John       | badminton  | sport centre a    | no     |
| 2           | 12-12-2020    | 13:00         | 1                 | watson     | swimming   | sport centre b    | no     |
| 3           | 12-12-2020    | 15:00         | 2                 | martin     | football   | sport centre c    | no     |
| 4           | 18-01-2021    | 09:00         | 8                 | mage       | volleyball | sport centre b    | no     |
| 5           | 22-01-2021    | 09:00         | 3                 | John       | badminton  | sport centre a    | no     |
| 6           | 23-01-2021    | 08:00         | 3                 | mage       | volleyball | sport centre b    | no     |
enter schedule id: 1
Succesfully modified record. Press enter to continue:
```

76. Registered student cancel class

```
MANAGE SCHEDULE
1. Book Class
2. Cancel Class
3. Return to Student Menu
enter choice: 2
| schedule_id | schedule_date | schedule_time | schedule_duration | coach_name | sport_name | sport_centre_name | booked |
-------------------------------------------------------------------------------------------------------------------
| 1           | 12-12-2020    | 09:00         | 4                 | John       | badminton  | sport centre a    | yes    |
| 2           | 12-12-2020    | 13:00         | 1                 | watson     | swimming   | sport centre b    | no     |
| 3           | 12-12-2020    | 15:00         | 2                 | martin     | football   | sport centre c    | no     |
| 4           | 18-01-2021    | 09:00         | 8                 | mage       | volleyball | sport centre b    | no     |
| 5           | 22-01-2021    | 09:00         | 3                 | John       | badminton  | sport centre a    | no     |
| 6           | 23-01-2021    | 08:00         | 3                 | mage       | volleyball | sport centre b    | no     |
enter schedule id: 1
Succesfully modified record. Press enter to continue:
```

77. Quit program



Additional Features

For this assignment, I added several additional features. One of the features I added was the ability to search for records using fields not specified in the assignment document. This includes searching with fields not from the same records, for example searching for coach records by the name of the sport they're teaching. Also, I added functionality to store passwords in their hashed forms, since storing passwords plaintext is bad for security.

I also added a function for registered students to book and cancel classes. Another feature I added is the ability to pick between sorting records in ascending order or in descending order. For admin and student logins, a max tries feature is added to limit the number of unsuccessful logins.

Conclusion

In conclusion, this system is working fine. Admins and students can use all the functionalities mentioned in the document, and several additional features to enhance their experience. Accessing and modifying records functionalities work. Logins and registrations also work.

References

1. TutorialsPoint (n.d.) "Python - Variable Types." Available at: https://www.tutorialspoint.com/python/python_variable_types.htm (accessed 22 January 2021).
2. TutorialsPoint (n.d.) "Python - List". Available at: https://www.tutorialspoint.com/python/python_lists.htm (accessed 22 January 2021).
3. TutorialsPoint (n.d.) "Python - Modules". Available at: https://www.tutorialspoint.com/python/python_modules.htm (accessed 22 January 2021).