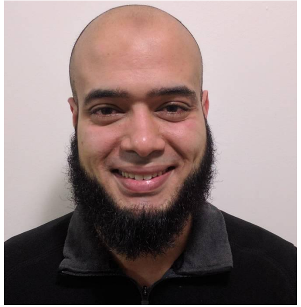
Programming4kids

Structures

Mostafa Saad Ibrahim

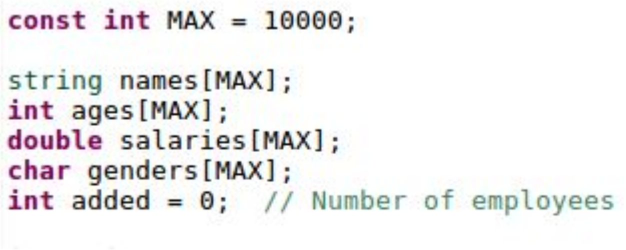
Computer Vision Researcher @ Huawei Canada
PhD - Simon Fraser University
Bachelor / Msc - FCI Cairo University

Ex-(Software Engineer / Teaching Assistant)



Recall the employee system

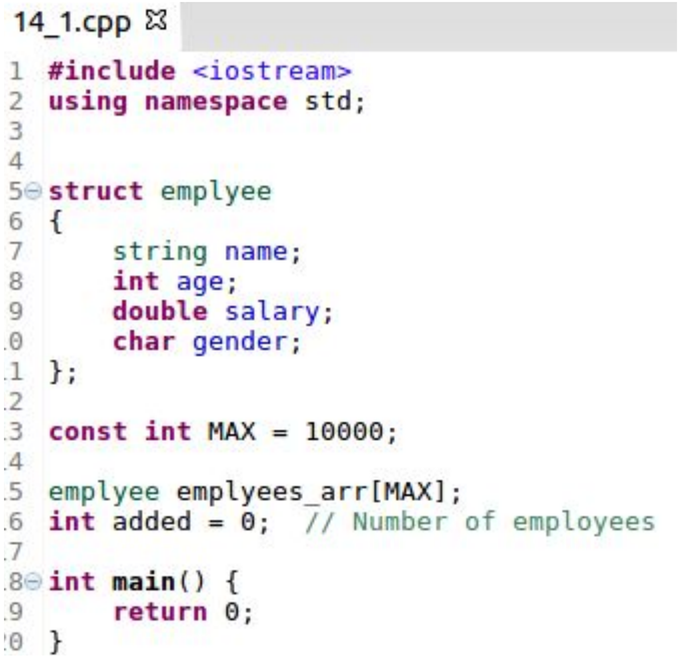
- Each Employee has
  - Name
  - Age
  - Salary
  - Gender
- So we defined 4 arrays
- In practice, this is hard to maintain
  - e.g. got new features: email address - home address
- C++ offers us a convenient way to solve this problem!



```
const int MAX = 10000;

string names[MAX];
int ages[MAX];
double salaries[MAX];
char genders[MAX];
int added = 0;  // Number of employees
```

Define a struct

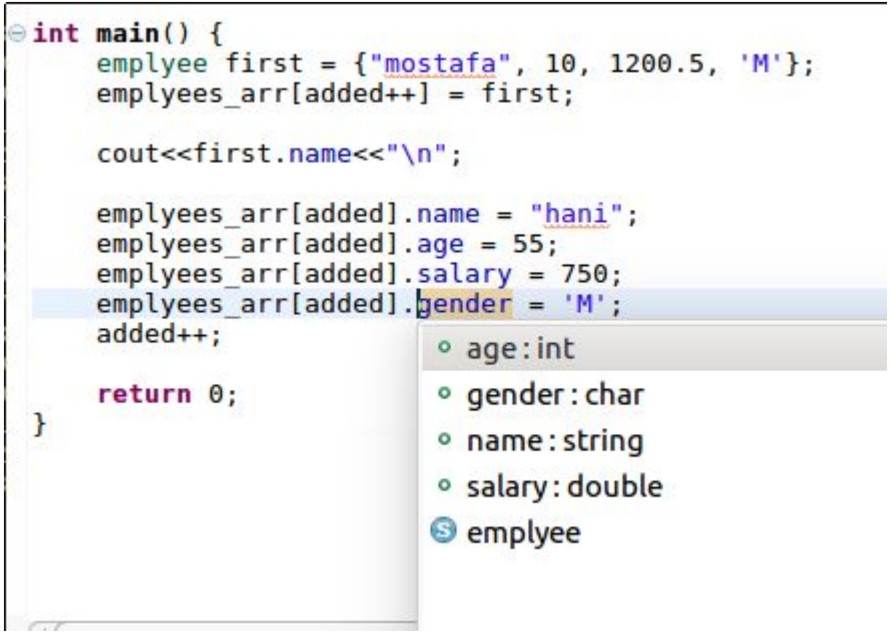
- Define a struct (e.g. data type)
- Add all elements inside it
- Create instance or array of it!
- Now, employee can change smoothly
- Also we defined 1 array only!



```cpp
#include <iostream>
using namespace std;


struct emplyee
{
    string name;
    int age;
    double salary;
    char gender;
};

const int MAX = 10000;

emplyee emplyees_arr[MAX];
int added = 0;  // Number of employees

int main() {
    return 0;
}
```

Creating elements

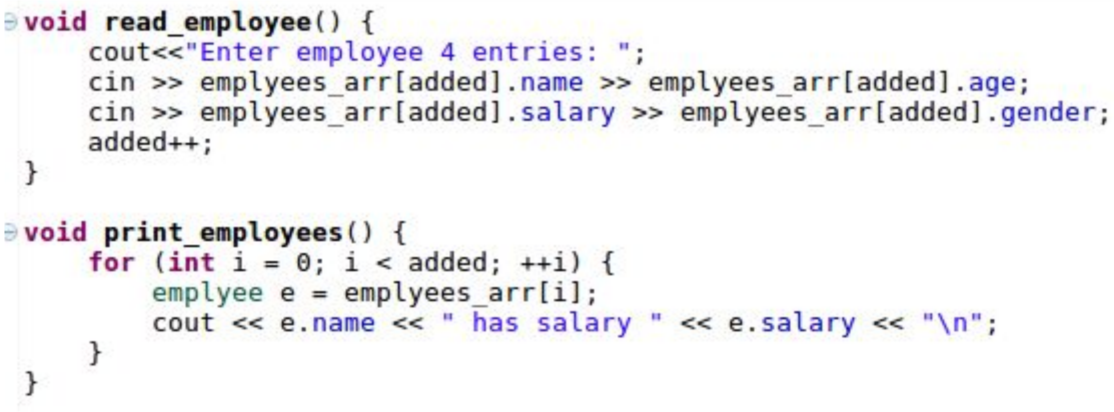
- In code, 2 ways to create instance
- Notice the dot to get elements!
- In IDE, right click + space
    - Given menu of choices!



```
int main() {
	emplyee first = {"mostafa", 10, 1200.5, 'M'};
	emplyees_arr[added++] = first;

	cout<<first.name<<"\n";

	emplyees_arr[added].name = "hani";
	emplyees_arr[added].age = 55;
	emplyees_arr[added].salary = 750;
	emplyees_arr[added].gender = 'M';
	added++;

	return 0;
}
```

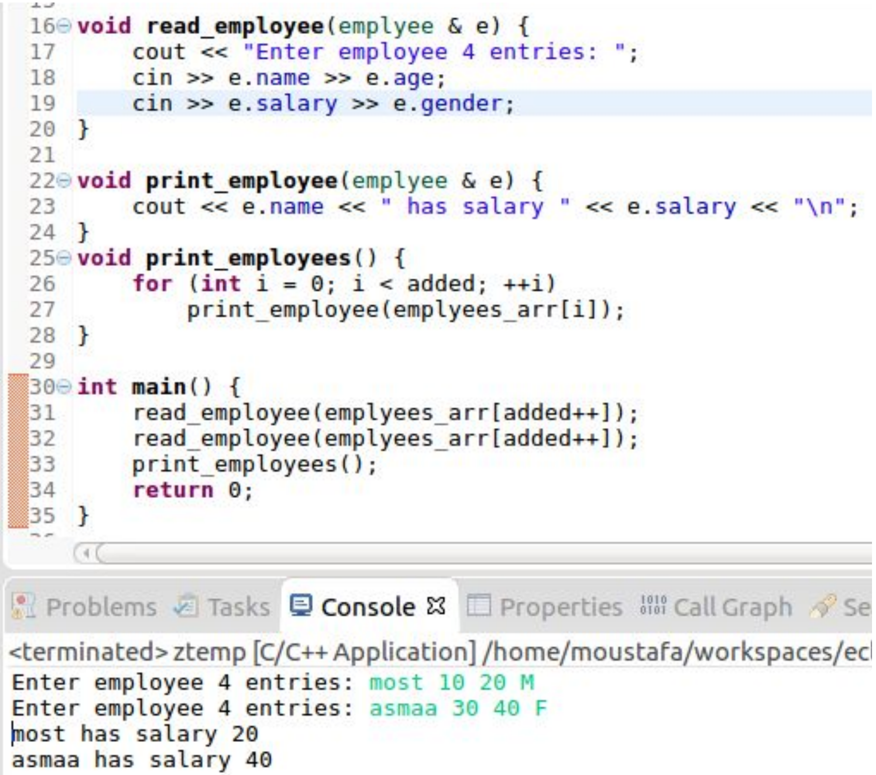
Reading & Writing

- Notice everything is now as we used to do
- You just access the variable using ‘.’ operator



```
void read_employee() {
    cout<<"Enter employee 4 entries: ";
    cin >> emplyees_arr[added].name >> emplyees_arr[added].age;
    cin >> emplyees_arr[added].salary >> emplyees_arr[added].gender;
    added++;
}

void print_employees() {
    for (int i = 0; i < added; ++i) {
        emplyee e = emplyees_arr[i];
        cout << e.name << " has salary " << e.salary << "\n";
    }
}
```

Reading & Writing - Another way

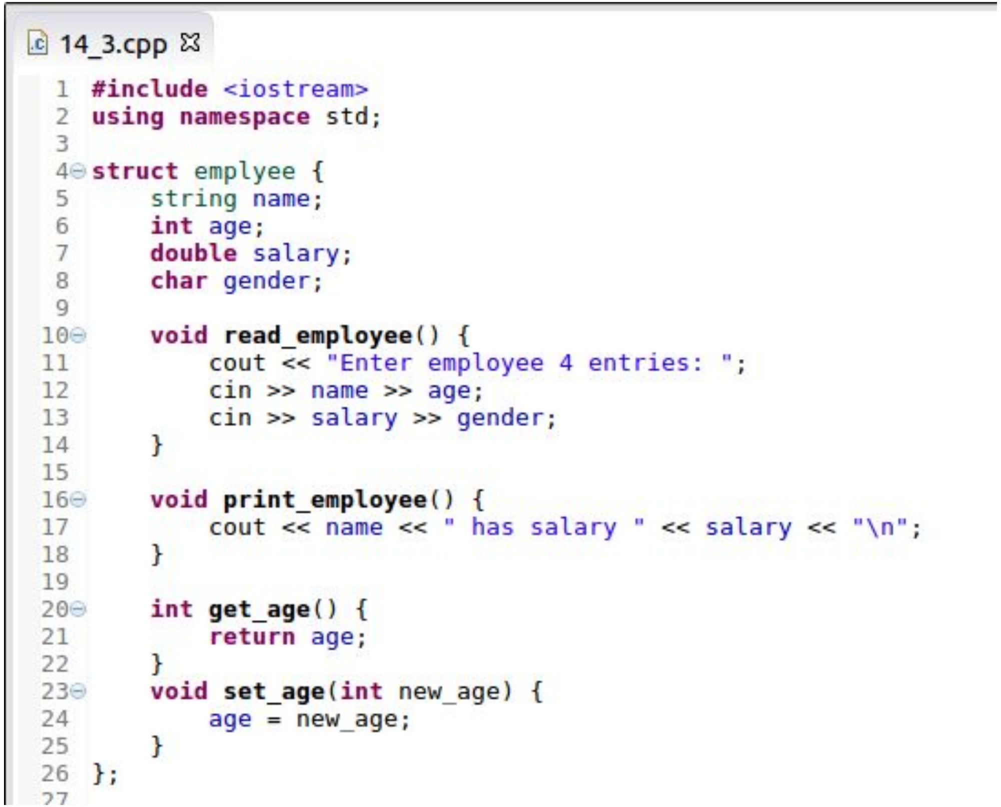


```cpp
void read_employee(emplyee & e) {
    cout << "Enter employee 4 entries: ";
    cin >> e.name >> e.age;
    cin >> e.salary >> e.gender;
}

void print_employee(emplyee & e) {
    cout << e.name << " has salary " << e.salary << "\n";
}
void print_employees() {
    for (int i = 0; i < added; ++i)
        print_employee(emplyees_arr[i]);
}

int main() {
    read_employee(emplyees_arr[added++]);
    read_employee(emplyees_arr[added++]);
    print_employees();
    return 0;
}
```

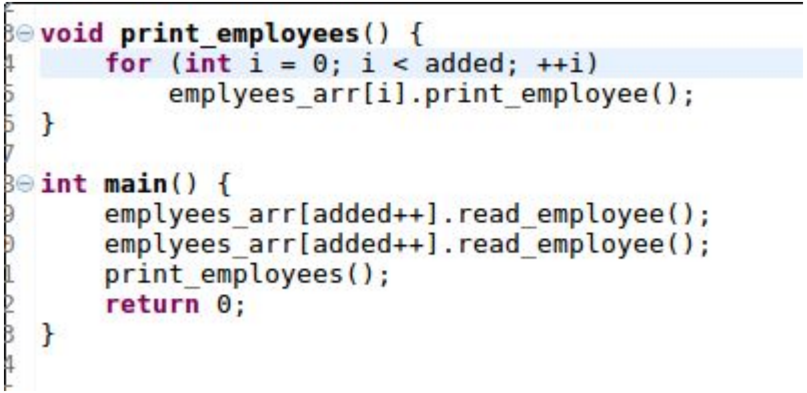
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/ec

```
Enter employee 4 entries: most 10 20 M
Enter employee 4 entries: asmaa 30 40 F
most has salary 20
asmaa has salary 40
```

Functions inside the struct



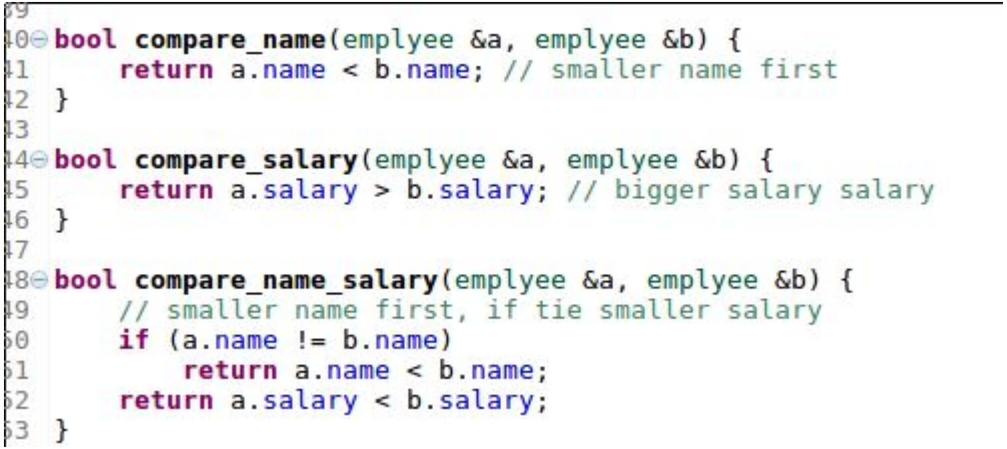
```cpp
#include <iostream>
using namespace std;

struct emplyee {
    string name;
    int age;
    double salary;
    char gender;

    void read_employee() {
        cout << "Enter employee 4 entries: ";
        cin >> name >> age;
        cin >> salary >> gender;
    }

    void print_employee() {
        cout << name << " has salary " << salary << "\n";
    }

    int get_age() {
        return age;
    }
    void set_age(int new_age) {
        age = new_age;
    }
};
```



```cpp
void print_employees() {
    for (int i = 0; i < added; ++i)
        emplyees_arr[i].print_employee();
}

int main() {
    emplyees_arr[added++].read_employee();
    emplyees_arr[added++].read_employee();
    print_employees();
    return 0;
}
```

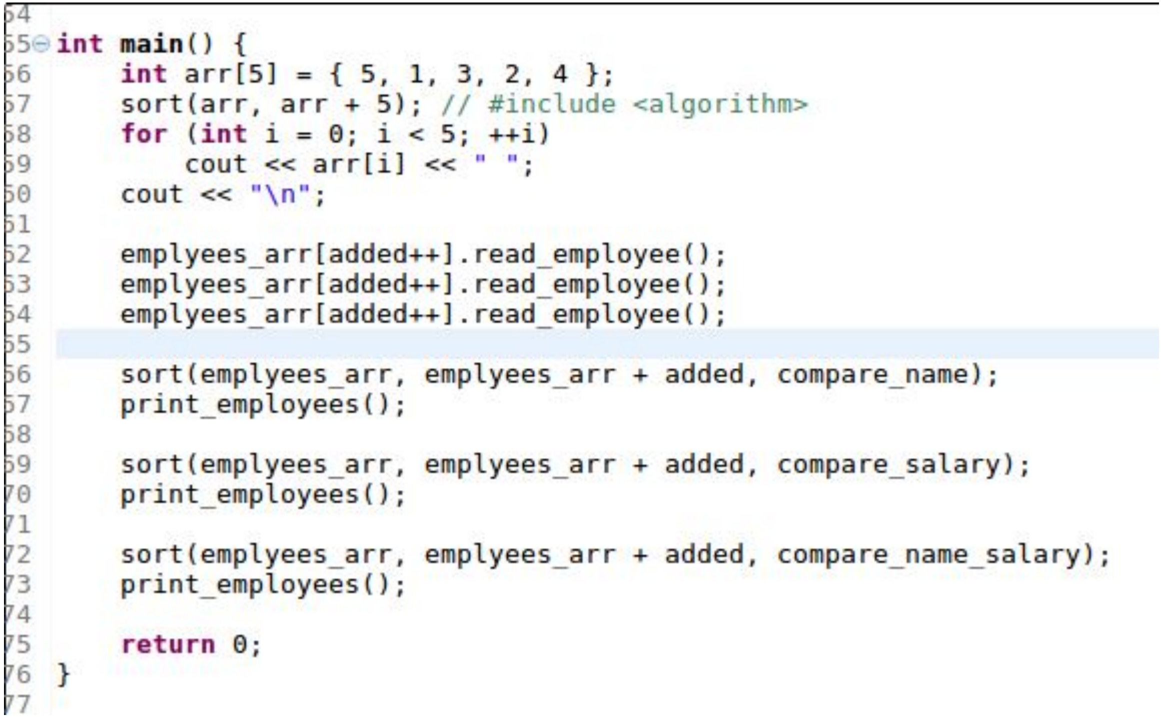
Compare Methods

- We can write functions that compare structs, normally
- Notice, if both sides has same value ⇒ false
    - E.g. if a.name = b.name = “ali”
- Always code it in this style

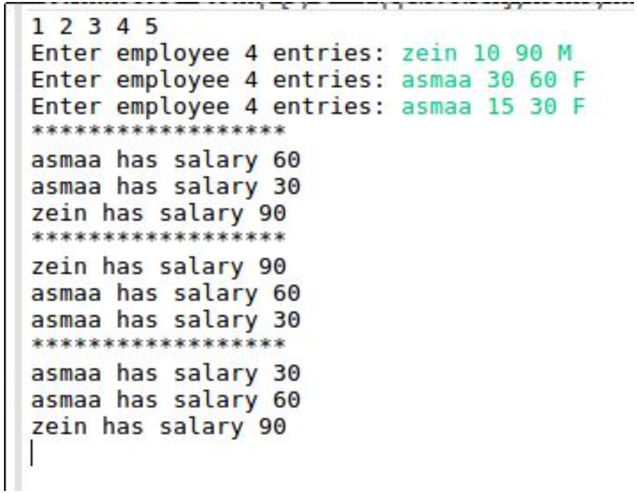


```
bool compare_name(emplyee &a, emplyee &b) {
    return a.name < b.name; // smaller name first
}

bool compare_salary(emplyee &a, emplyee &b) {
    return a.salary > b.salary; // bigger salary salary
}

bool compare_name_salary(emplyee &a, emplyee &b) {
    // smaller name first, if tie smaller salary
    if (a.name != b.name)
        return a.name < b.name;
    return a.salary < b.salary;
}
```

Ordering array

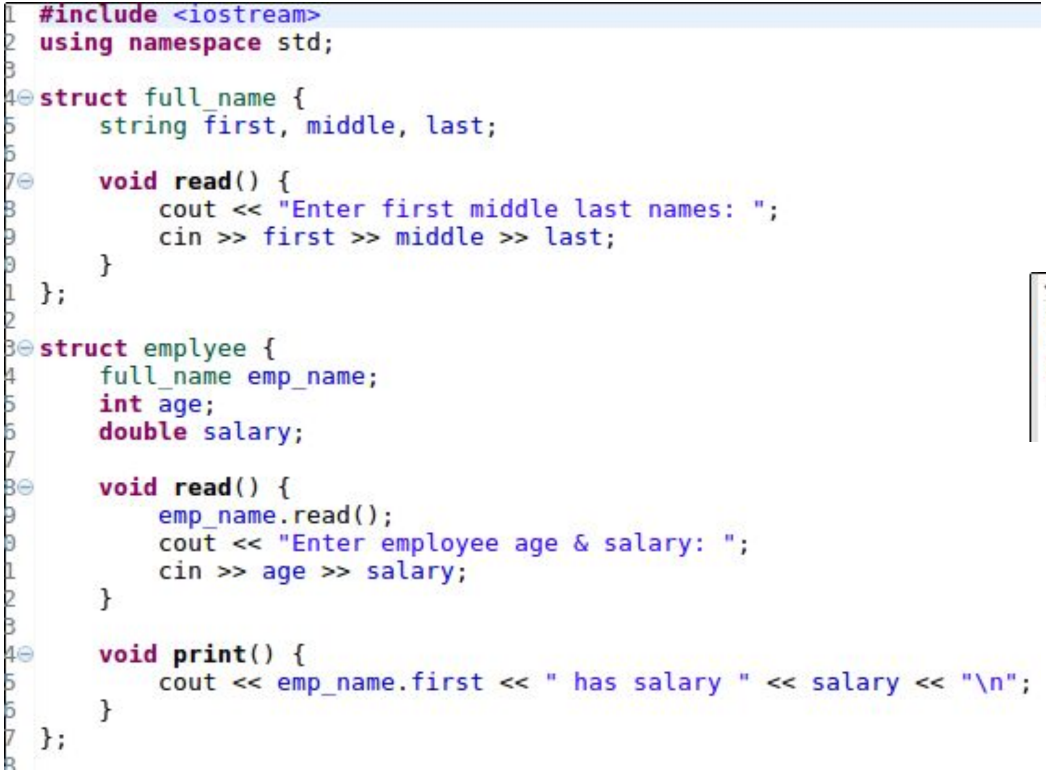


```
int main() {
    int arr[5] = { 5, 1, 3, 2, 4 };
    sort(arr, arr + 5); // #include <algorithm>
    for (int i = 0; i < 5; ++i)
        cout << arr[i] << " ";
    cout << "\n";

    emplyees_arr[added++].read_employee();
    emplyees_arr[added++].read_employee();
    emplyees_arr[added++].read_employee();

    sort(emplyees_arr, emplyees_arr + added, compare_name);
    print_employees();

    sort(emplyees_arr, emplyees_arr + added, compare_salary);
    print_employees();

    sort(emplyees_arr, emplyees_arr + added, compare_name_salary);
    print_employees();

    return 0;
}
```

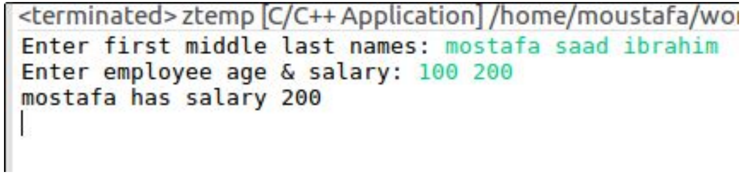


```
1 2 3 4 5
Enter employee 4 entries: zein 10 90 M
Enter employee 4 entries: asmaa 30 60 F
Enter employee 4 entries: asmaa 15 30 F
******************
asmaa has salary 60
asmaa has salary 30
zein has salary 90
******************
zein has salary 90
asmaa has salary 60
asmaa has salary 30
******************
asmaa has salary 30
asmaa has salary 60
zein has salary 90
```

Structure of structure

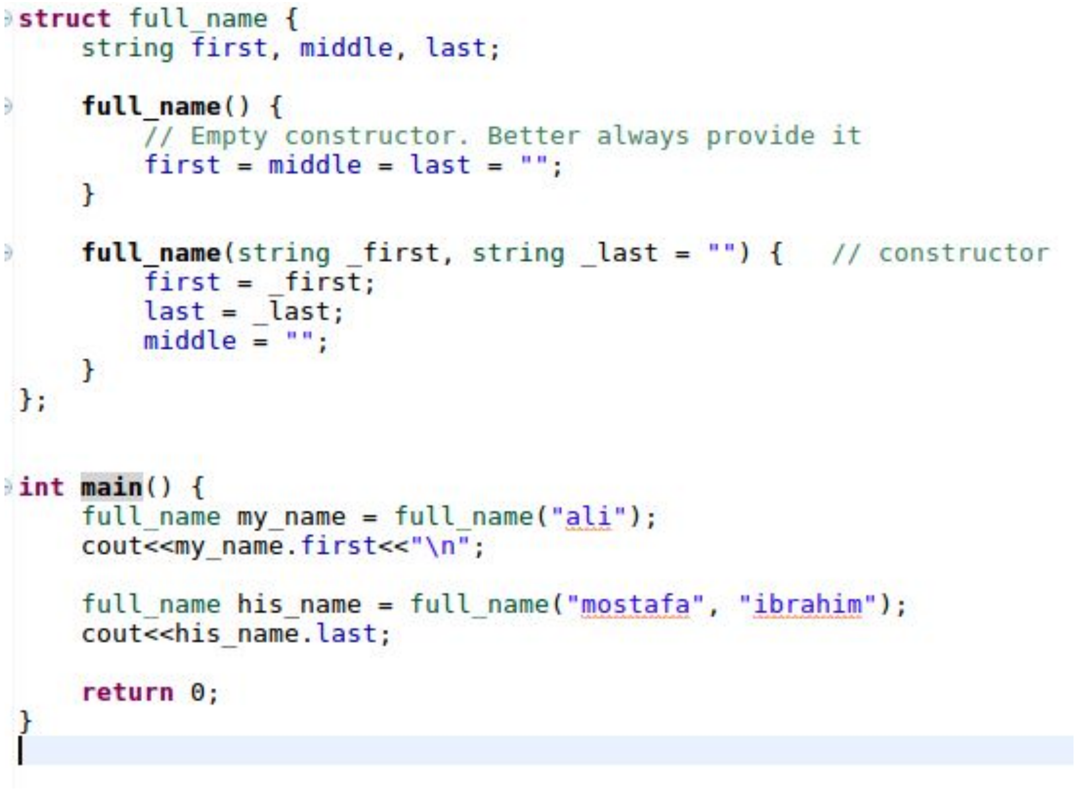


```cpp
#include <iostream>
using namespace std;

struct full_name {
    string first, middle, last;

    void read() {
        cout << "Enter first middle last names: ";
        cin >> first >> middle >> last;
    }
};

struct emplyee {
    full_name emp_name;
    int age;
    double salary;

    void read() {
        emp_name.read();
        cout << "Enter employee age & salary: ";
        cin >> age >> salary;
    }

    void print() {
        cout << emp_name.first << " has salary " << salary << "\n";
    }
};
```



```
Enter first middle last names: mostafa saad ibrahim
Enter employee age & salary: 100 200
mostafa has salary 200
```

Constructor

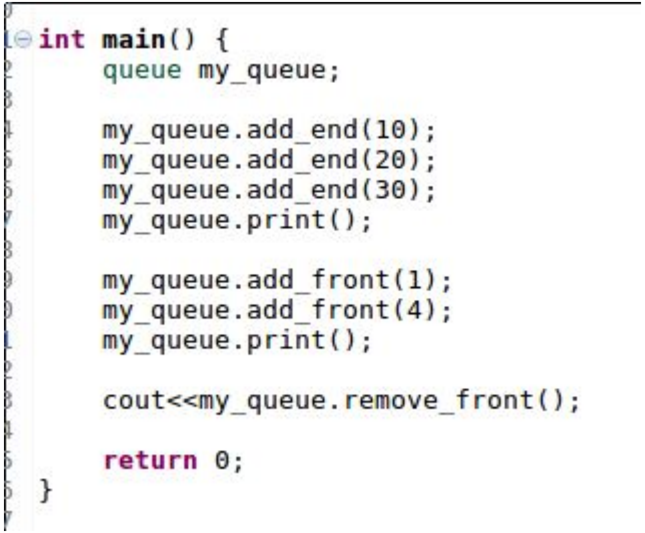


```
struct full_name {
    string first, middle, last;

    full_name() {
        // Empty constructor. Better always provide it
        first = middle = last = "";
    }

    full_name(string _first, string _last = "") {   // constructor
        first = _first;
        last = _last;
        middle = "";
    }
};


int main() {
    full_name my_name = full_name("ali");
    cout<<my_name.first<<"\n";

    full_name his_name = full_name("mostafa", "ibrahim");
    cout<<his_name.last;

    return 0;
}
```

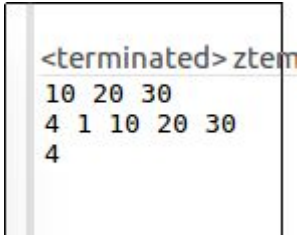
Practice: Our own queue

- Define a class, name it queue. It should internally have an array and support following operations
    - void add_end(int value): add to the end of current array
    - void add_front(int value): add to the front of this array
    - int remove_front(): remove the front value and remove it. Return the value
    - void print(): print the array

Practice: Our own queue

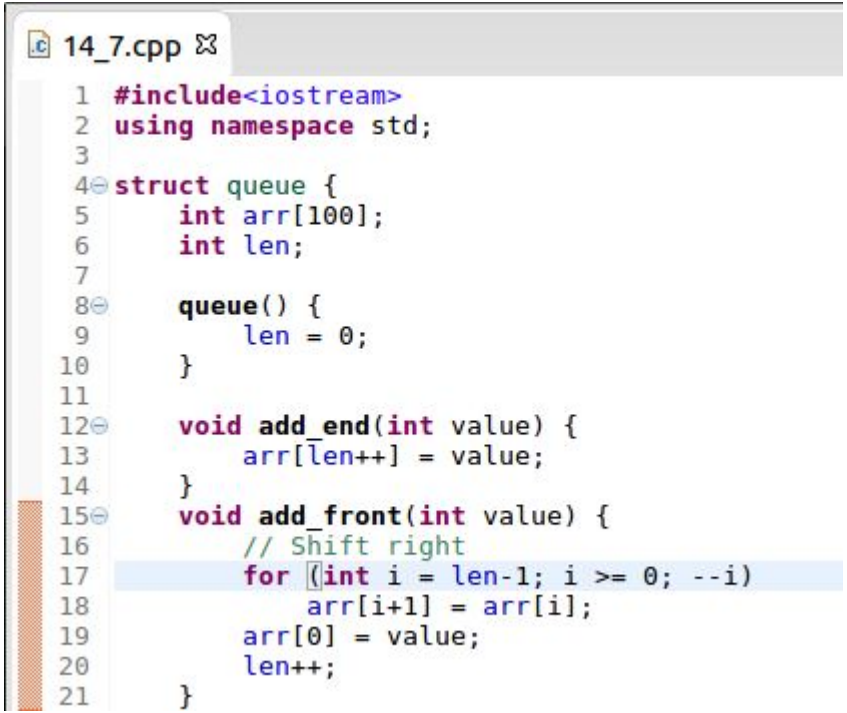


```
int main() {
    queue my_queue;

    my_queue.add_end(10);
    my_queue.add_end(20);
    my_queue.add_end(30);
    my_queue.print();

    my_queue.add_front(1);
    my_queue.add_front(4);
    my_queue.print();

    cout<<my_queue.remove_front();

    return 0;
}
```



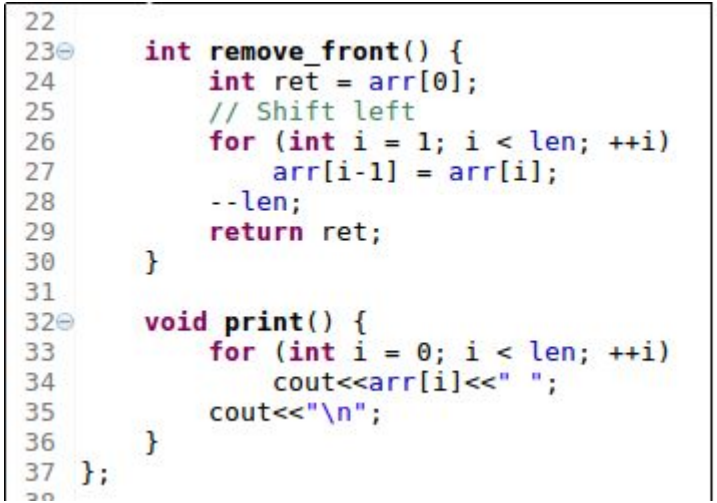
```
10 20 30
4 1 10 20 30
4
```

Practice: Our own queue



14_7.cpp

```cpp
#include<iostream>
using namespace std;

struct queue {
    int arr[100];
    int len;

    queue() {
        len = 0;
    }

    void add_end(int value) {
        arr[len++] = value;
    }
    void add_front(int value) {
        // Shift right
        for (int i = len-1; i >= 0; --i)
            arr[i+1] = arr[i];
        arr[0] = value;
        len++;
    }
```



```cpp

    int remove_front() {
        int ret = arr[0];
        // Shift left
        for (int i = 1; i < len; ++i)
            arr[i-1] = arr[i];
        --len;
        return ret;
    }

    void print() {
        for (int i = 0; i < len; ++i)
            cout<<arr[i]<<" ";
        cout<<"\n";
    }
};
```

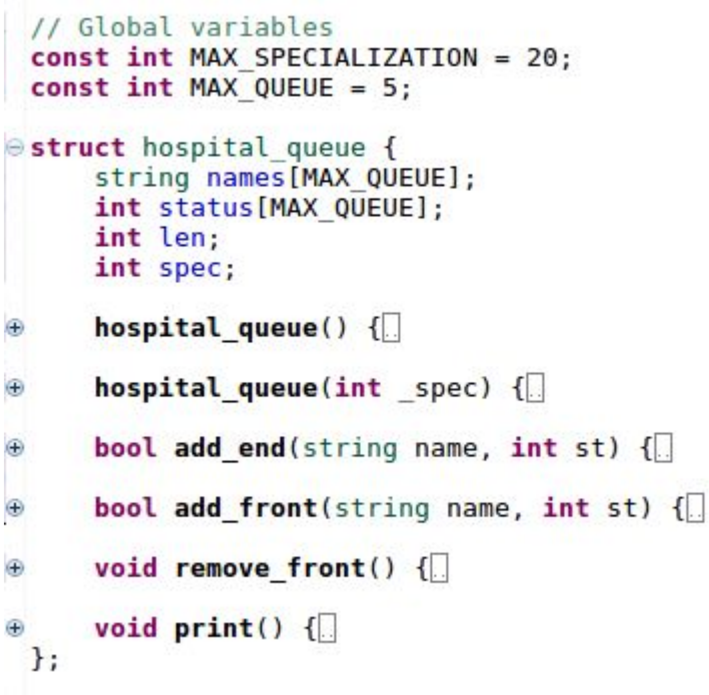
Practice - Hospital System

- Let’s rewrite again the last hospital system
- Give yourself ‘good trial’ in thinking how to re-write

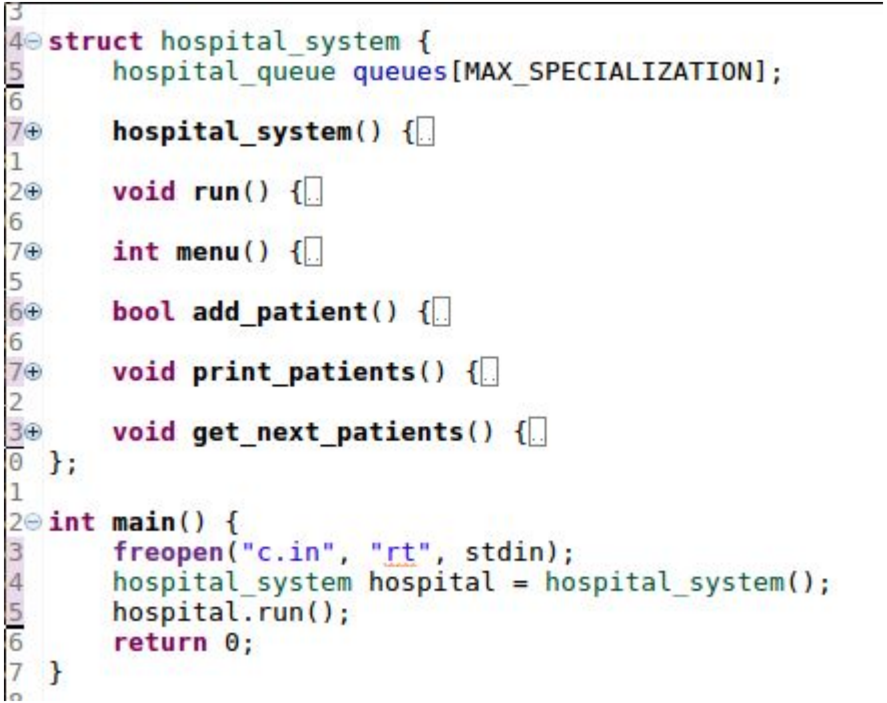
Practice - Hospital System

- Let’s create hospital_queue, like the previous queue
  - Variables
  - string names[MAX_QUEUE];
  - int status[MAX_QUEUE];
  - int len;
  - int spec;
  - Provide same functionalities
- Let’s create hospital_system
  - hospital_queue queues[MAX_SPECIALIZATION];
  - Add the methods inside it using the hospital_queue change

Practice - Hospital System - Big Picture

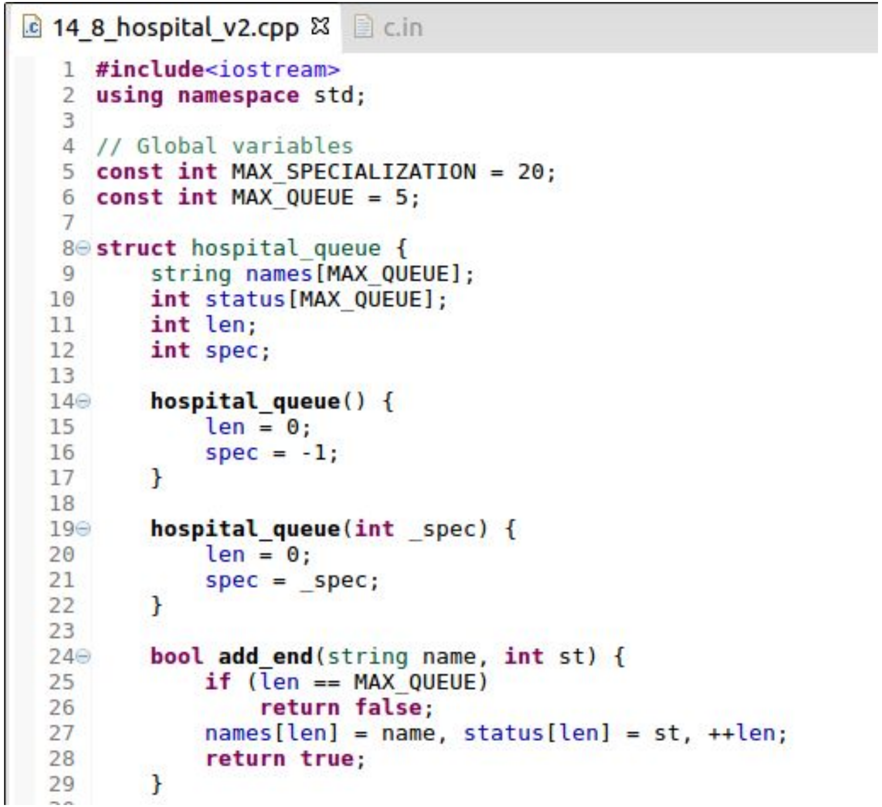


```
// Global variables
const int MAX_SPECIALIZATION = 20;
const int MAX_QUEUE = 5;

struct hospital_queue {
    string names[MAX_QUEUE];
    int status[MAX_QUEUE];
    int len;
    int spec;

    hospital_queue() {□

    hospital_queue(int _spec) {□

    bool add_end(string name, int st) {□

    bool add_front(string name, int st) {□

    void remove_front() {□

    void print() {□
};
```



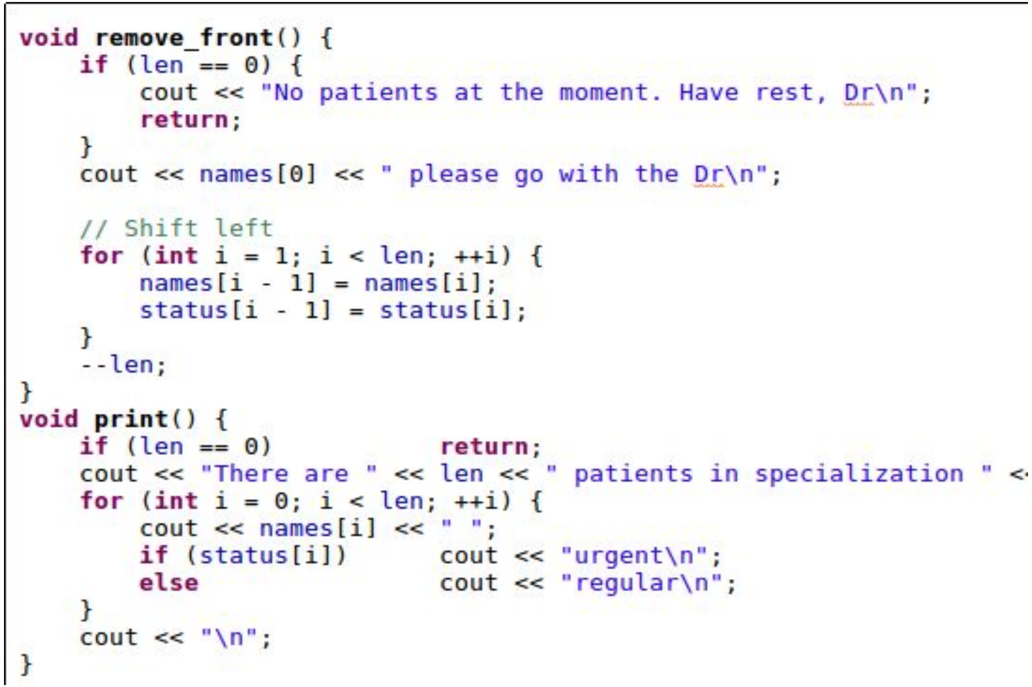
```
struct hospital_system {
    hospital_queue queues[MAX_SPECIALIZATION];

    hospital_system() {□

    void run() {□

    int menu() {□

    bool add_patient() {□

    void print_patients() {□

    void get_next_patients() {□
};

int main() {
    freopen("c.in", "rt", stdin);
    hospital_system hospital = hospital_system();
    hospital.run();
    return 0;
}
```

Practice - Hospital System



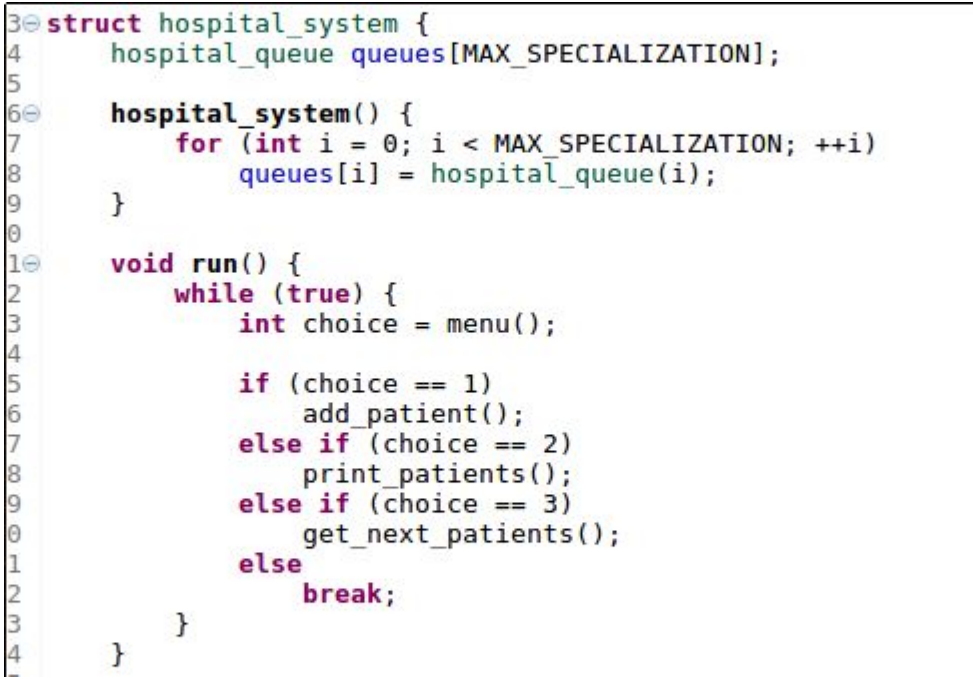
```cpp
#include<iostream>
using namespace std;

// Global variables
const int MAX_SPECIALIZATION = 20;
const int MAX_QUEUE = 5;

struct hospital_queue {
    string names[MAX_QUEUE];
    int status[MAX_QUEUE];
    int len;
    int spec;

    hospital_queue() {
        len = 0;
        spec = -1;
    }

    hospital_queue(int _spec) {
        len = 0;
        spec = _spec;
    }

    bool add_end(string name, int st) {
        if (len == MAX_QUEUE)
            return false;
        names[len] = name, status[len] = st, ++len;
        return true;
    }
```

Practice - Hospital System

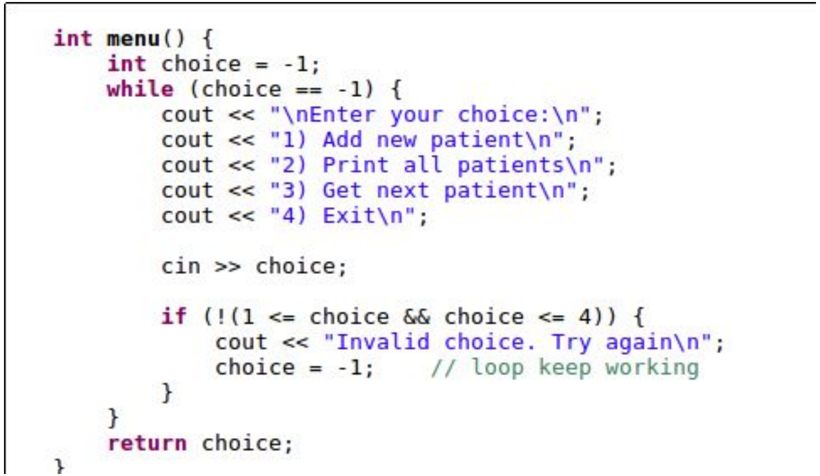


```
void remove_front() {
    if (len == 0) {
        cout << "No patients at the moment. Have rest, Dr\n";
        return;
    }
    cout << names[0] << " please go with the Dr\n";

    // Shift left
    for (int i = 1; i < len; ++i) {
        names[i - 1] = names[i];
        status[i - 1] = status[i];
    }
    --len;
}
void print() {
    if (len == 0)           return;
    cout << "There are " << len << " patients in specialization " <<
    for (int i = 0; i < len; ++i) {
        cout << names[i] << " ";
        if (status[i])      cout << "urgent\n";
        else                cout << "regular\n";
    }
    cout << "\n";
}
```

Practice - Hospital System

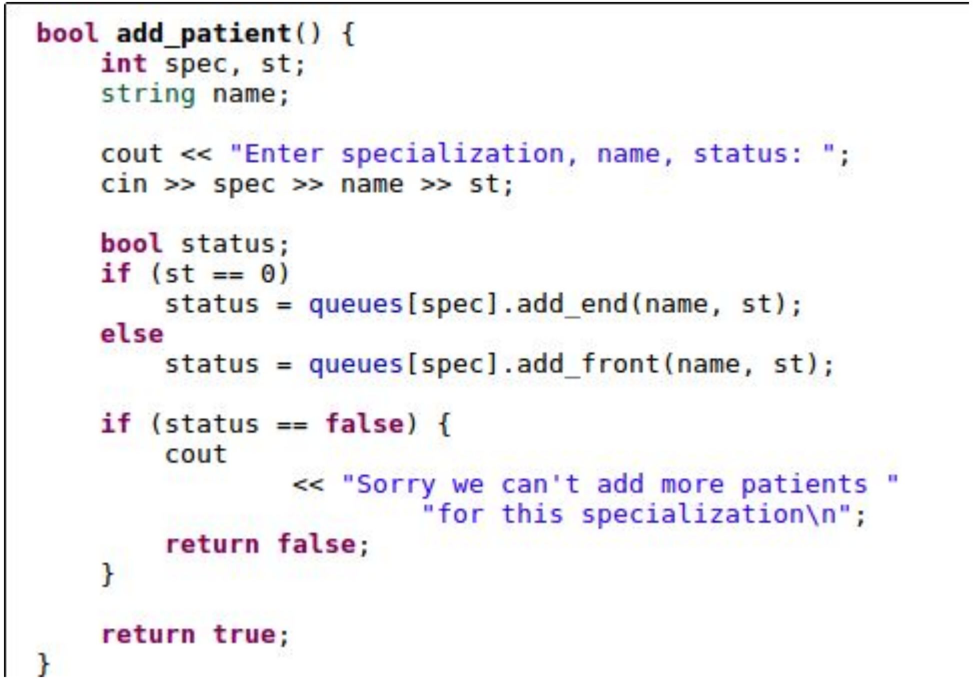


```
struct hospital_system {
    hospital_queue queues[MAX_SPECIALIZATION];

    hospital_system() {
        for (int i = 0; i < MAX_SPECIALIZATION; ++i)
            queues[i] = hospital_queue(i);
    }

    void run() {
        while (true) {
            int choice = menu();

            if (choice == 1)
                add_patient();
            else if (choice == 2)
                print_patients();
            else if (choice == 3)
                get_next_patients();
            else
                break;

        }
    }
```



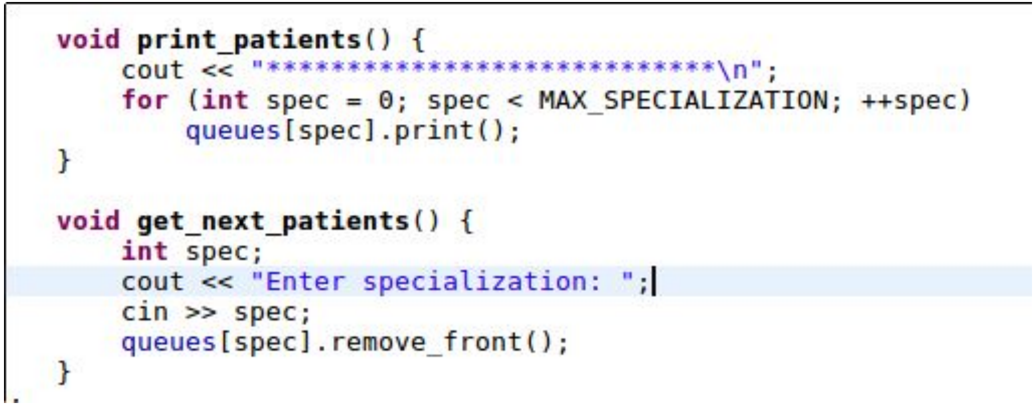
```
int menu() {
    int choice = -1;
    while (choice == -1) {
        cout << "\nEnter your choice:\n";
        cout << "1) Add new patient\n";
        cout << "2) Print all patients\n";
        cout << "3) Get next patient\n";
        cout << "4) Exit\n";

        cin >> choice;

        if (!(1 <= choice && choice <= 4)) {
            cout << "Invalid choice. Try again\n";
            choice = -1;    // loop keep working
        }
    }
    return choice;
}
```

Practice - Hospital System



```cpp
bool add_patient() {
    int spec, st;
    string name;

    cout << "Enter specialization, name, status: ";
    cin >> spec >> name >> st;

    bool status;
    if (st == 0)
        status = queues[spec].add_end(name, st);
    else
        status = queues[spec].add_front(name, st);

    if (status == false) {
        cout
                << "Sorry we can't add more patients "
                        "for this specialization\n";
        return false;
    }

    return true;
}
```

Practice - Hospital System



```
void print_patients() {
    cout << "****************************\n";
    for (int spec = 0; spec < MAX_SPECIALIZATION; ++spec)
        queues[spec].print();
}

void get_next_patients() {
    int spec;
    cout << "Enter specialization: ";
    cin >> spec;
    queues[spec].remove_front();
}
```

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً

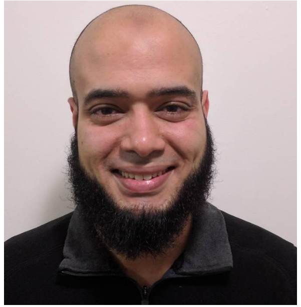
Programming4kids

Structures

Mostafa Saad Ibrahim

Computer Vision Researcher @ Huawei Canada
PhD - Simon Fraser University
Bachelor / Msc - FCI Cairo University

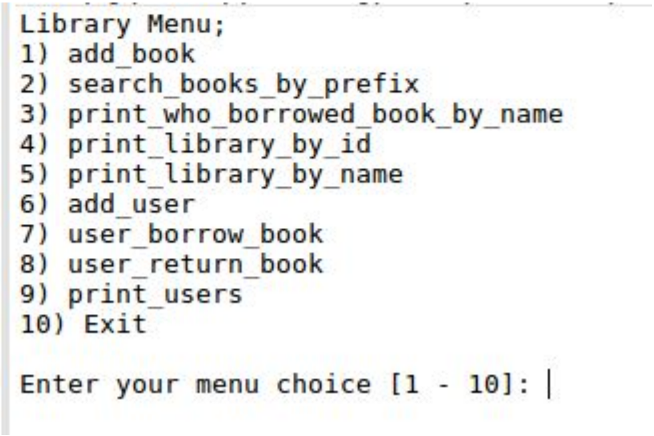
Ex-(Software Engineer / Teaching Assistant)



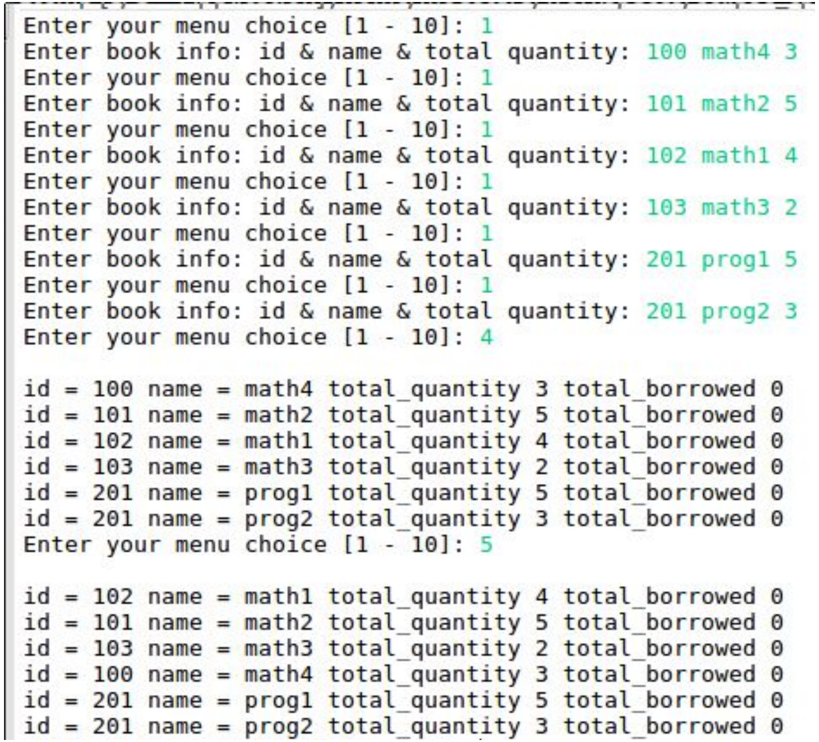
Homework: Library System

- Create a library system to maintain information about books and users
- Each book has: id (int) - name - quantity
- Each user has: id - name
- System allows adding a book or a user by reading the above information
- Books operations
    - Given a prefix, list all books with this prefix string
    - List the books sorted either based on id or name
    - Given a book name, list all users borrowed such a book
- User operations
    - User can borrow or return a book
    - List all system users: With borrowed books **IDs sorted**

Homework: Library System

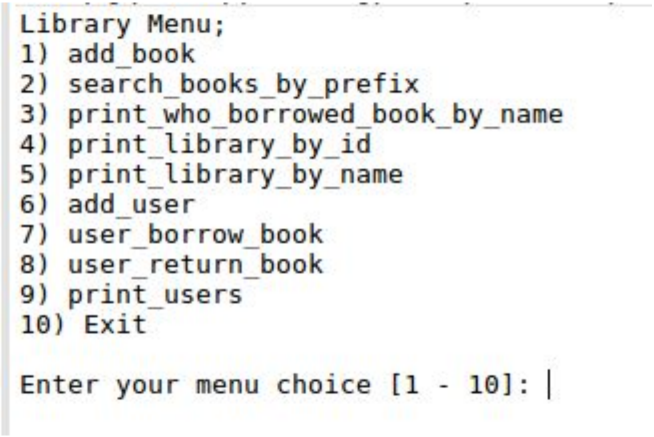


```
Library Menu;
1) add_book
2) search_books_by_prefix
3) print_who_borrowed_book_by_name
4) print_library_by_id
5) print_library_by_name
6) add_user
7) user_borrow_book
8) user_return_book
9) print_users
10) Exit

Enter your menu choice [1 - 10]: |
```

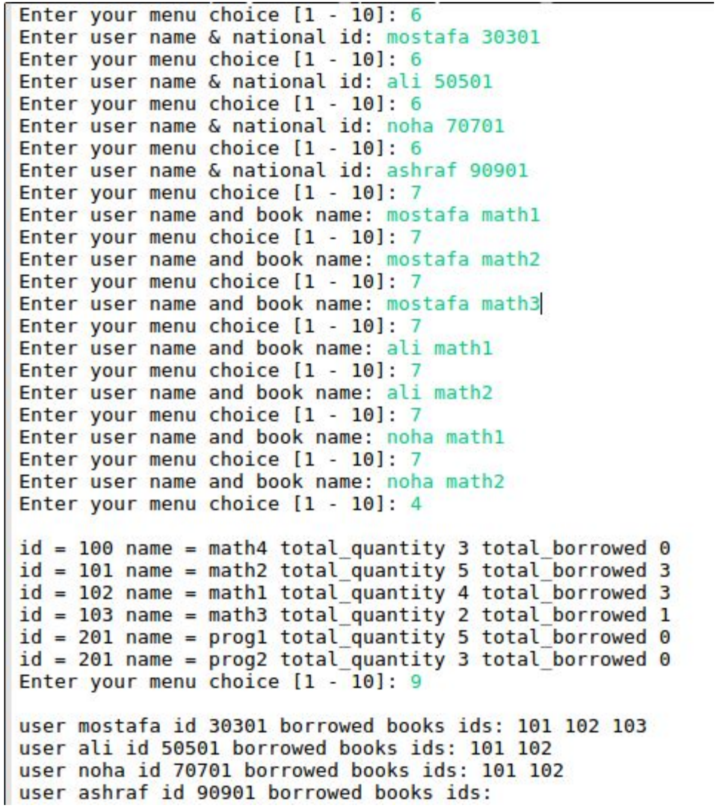


```
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 100 math4 3
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 101 math2 5
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 102 math1 4
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 103 math3 2
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 201 prog1 5
Enter your menu choice [1 - 10]: 1
Enter book info: id & name & total quantity: 201 prog2 3
Enter your menu choice [1 - 10]: 4

id = 100 name = math4 total_quantity 3 total_borrowed 0
id = 101 name = math2 total_quantity 5 total_borrowed 0
id = 102 name = math1 total_quantity 4 total_borrowed 0
id = 103 name = math3 total_quantity 2 total_borrowed 0
id = 201 name = prog1 total_quantity 5 total_borrowed 0
id = 201 name = prog2 total_quantity 3 total_borrowed 0
Enter your menu choice [1 - 10]: 5

id = 102 name = math1 total_quantity 4 total_borrowed 0
id = 101 name = math2 total_quantity 5 total_borrowed 0
id = 103 name = math3 total_quantity 2 total_borrowed 0
id = 100 name = math4 total_quantity 3 total_borrowed 0
id = 201 name = prog1 total_quantity 5 total_borrowed 0
id = 201 name = prog2 total_quantity 3 total_borrowed 0
```

Homework: Library System

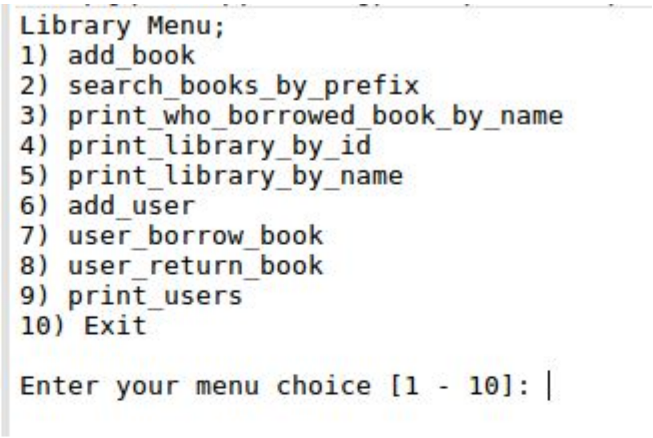


```
Library Menu;
1) add_book
2) search_books_by_prefix
3) print_who_borrowed_book_by_name
4) print_library_by_id
5) print_library_by_name
6) add_user
7) user_borrow_book
8) user_return_book
9) print_users
10) Exit

Enter your menu choice [1 - 10]: |
```

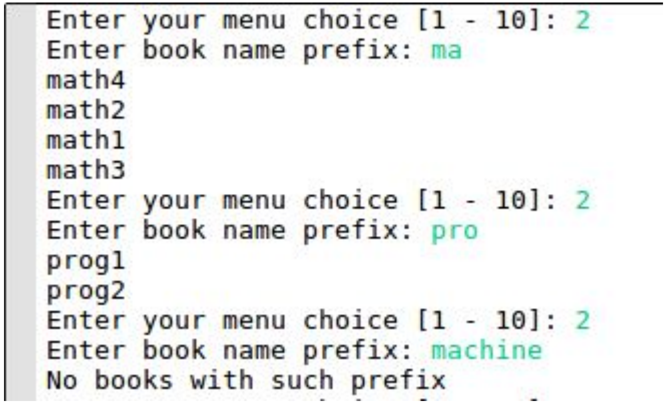


```
Enter your menu choice [1 - 10]: 6
Enter user name & national id: mostafa 30301
Enter your menu choice [1 - 10]: 6
Enter user name & national id: ali 50501
Enter your menu choice [1 - 10]: 6
Enter user name & national id: noha 70701
Enter your menu choice [1 - 10]: 6
Enter user name & national id: ashraf 90901
Enter your menu choice [1 - 10]: 7
Enter user name and book name: mostafa math1
Enter your menu choice [1 - 10]: 7
Enter user name and book name: mostafa math2
Enter your menu choice [1 - 10]: 7
Enter user name and book name: mostafa math3
Enter your menu choice [1 - 10]: 7
Enter user name and book name: ali math1
Enter your menu choice [1 - 10]: 7
Enter user name and book name: ali math2
Enter your menu choice [1 - 10]: 7
Enter user name and book name: noha math1
Enter your menu choice [1 - 10]: 7
Enter user name and book name: noha math2
Enter your menu choice [1 - 10]: 4

id = 100 name = math4 total_quantity 3 total_borrowed 0
id = 101 name = math2 total_quantity 5 total_borrowed 3
id = 102 name = math1 total_quantity 4 total_borrowed 3
id = 103 name = math3 total_quantity 2 total_borrowed 1
id = 201 name = prog1 total_quantity 5 total_borrowed 0
id = 201 name = prog2 total_quantity 3 total_borrowed 0
Enter your menu choice [1 - 10]: 9

user mostafa id 30301 borrowed books ids: 101 102 103
user ali id 50501 borrowed books ids: 101 102
user noha id 70701 borrowed books ids: 101 102
user ashraf id 90901 borrowed books ids:
```

Homework: Library System

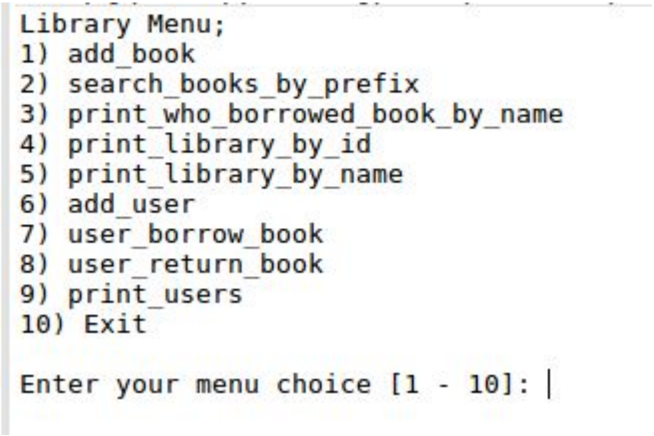


```
Library Menu;
1) add_book
2) search_books_by_prefix
3) print_who_borrowed_book_by_name
4) print_library_by_id
5) print_library_by_name
6) add_user
7) user_borrow_book
8) user_return_book
9) print_users
10) Exit

Enter your menu choice [1 - 10]: |
```

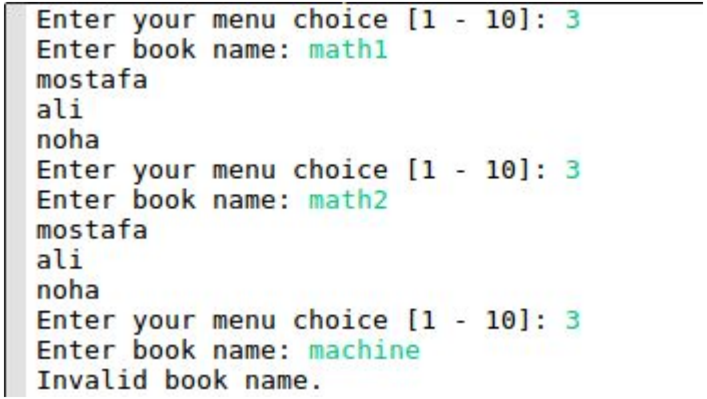


```
Enter your menu choice [1 - 10]: 2
Enter book name prefix: ma
math4
math2
math1
math3
Enter your menu choice [1 - 10]: 2
Enter book name prefix: pro
prog1
prog2
Enter your menu choice [1 - 10]: 2
Enter book name prefix: machine
No books with such prefix
```

Homework: Library System

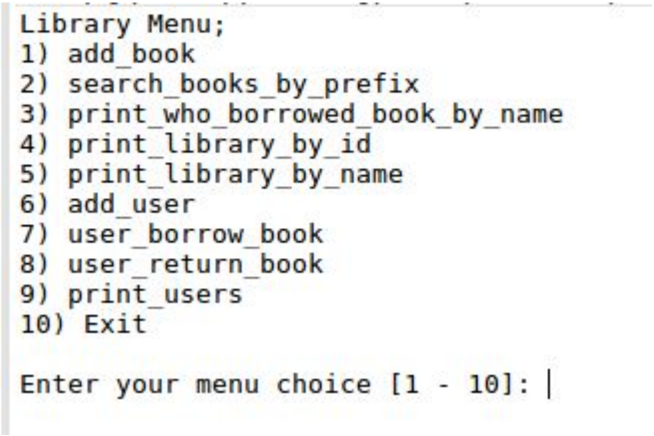


```
Library Menu;
1) add_book
2) search_books_by_prefix
3) print_who_borrowed_book_by_name
4) print_library_by_id
5) print_library_by_name
6) add_user
7) user_borrow_book
8) user_return_book
9) print_users
10) Exit

Enter your menu choice [1 - 10]: 
```

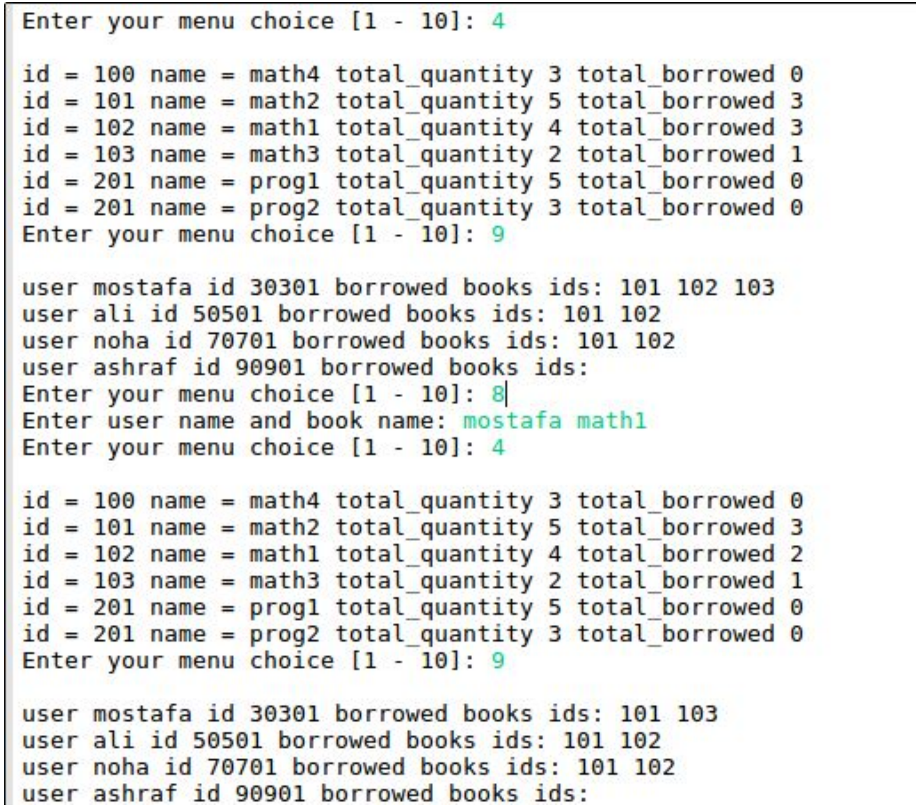


```
Enter your menu choice [1 - 10]: 3
Enter book name: math1
mostafa
ali
noha
Enter your menu choice [1 - 10]: 3
Enter book name: math2
mostafa
ali
noha
Enter your menu choice [1 - 10]: 3
Enter book name: machine
Invalid book name.
```

Homework: Library System



```
Library Menu;
1) add_book
2) search_books_by_prefix
3) print_who_borrowed_book_by_name
4) print_library_by_id
5) print_library_by_name
6) add_user
7) user_borrow_book
8) user_return_book
9) print_users
10) Exit

Enter your menu choice [1 - 10]:
```



```
Enter your menu choice [1 - 10]: 4

id = 100 name = math4 total_quantity 3 total_borrowed 0
id = 101 name = math2 total_quantity 5 total_borrowed 3
id = 102 name = math1 total_quantity 4 total_borrowed 3
id = 103 name = math3 total_quantity 2 total_borrowed 1
id = 201 name = prog1 total_quantity 5 total_borrowed 0
id = 201 name = prog2 total_quantity 3 total_borrowed 0
Enter your menu choice [1 - 10]: 9

user mostafa id 30301 borrowed books ids: 101 102 103
user ali id 50501 borrowed books ids: 101 102
user noha id 70701 borrowed books ids: 101 102
user ashraf id 90901 borrowed books ids:
Enter your menu choice [1 - 10]: 8
Enter user name and book name: mostafa math1
Enter your menu choice [1 - 10]: 4

id = 100 name = math4 total_quantity 3 total_borrowed 0
id = 101 name = math2 total_quantity 5 total_borrowed 3
id = 102 name = math1 total_quantity 4 total_borrowed 2
id = 103 name = math3 total_quantity 2 total_borrowed 1
id = 201 name = prog1 total_quantity 5 total_borrowed 0
id = 201 name = prog2 total_quantity 3 total_borrowed 0
Enter your menu choice [1 - 10]: 9

user mostafa id 30301 borrowed books ids: 101 103
user ali id 50501 borrowed books ids: 101 102
user noha id 70701 borrowed books ids: 101 102
user ashraf id 90901 borrowed books ids:
```

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً