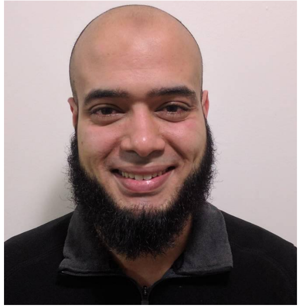
Programming4kids

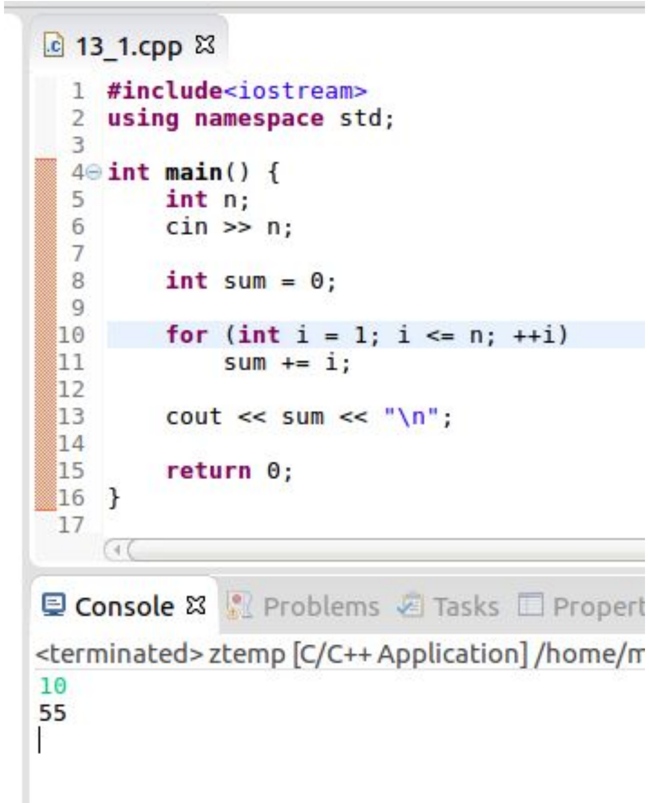
Functions

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

Ex-(Software Engineer / Teaching Assistant)



Users wanna sum from 1 to n

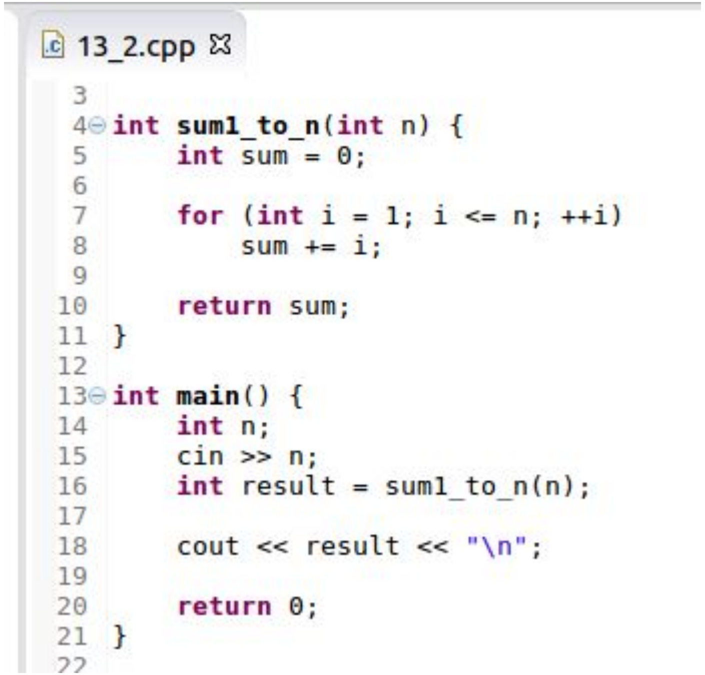


```cpp
#include<iostream>
using namespace std;

int main() {
    int n;
    cin >> n;

    int sum = 0;

    for (int i = 1; i <= n; ++i)
        sum += i;

    cout << sum << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/m
10
55
```

- Easy to write
- Very repetitive task!
- Now we want to give it to programmers like you
- How to do so!
- Functions

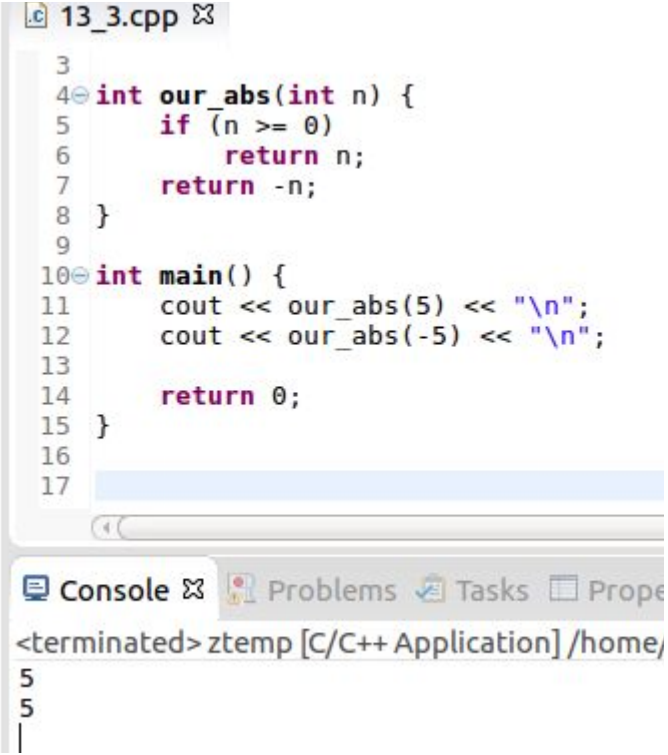
Function: sum1_to_n



```
 3
 4 int sum1_to_n(int n) {
 5     int sum = 0;
 6
 7     for (int i = 1; i <= n; ++i)
 8         sum += i;
 9
10     return sum;
11 }
12
13 int main() {
14     int n;
15     cin >> n;
16     int result = sum1_to_n(n);
17
18     cout << result << "\n";
19
20     return 0;
21 }
22
```

- int sum1_to_n(int n)
  - int = compute and return int
  - Sum1_to_n = function name
  - int n = parameter
- int result = sum1_to_n(n);
  - Sum1_to_n(n) = call it
  - int result = expect integer result
- Scope
  - In line 17 we can’t see variable sum in line 5

Function: Absolute

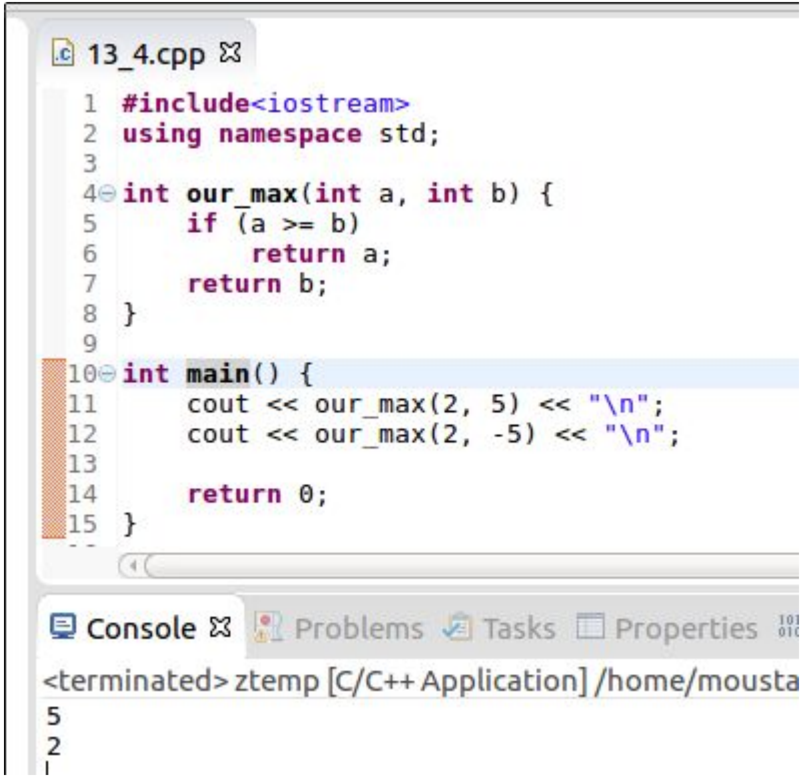


```cpp

int our_abs(int n) {
    if (n >= 0)
        return n;
    return -n;
}

int main() {
    cout << our_abs(5) << "\n";
    cout << our_abs(-5) << "\n";

    return 0;
}


```

```
<terminated> ztemp [C/C++ Application] /home/
5
5
```

- Absolute is a function that return positive value for the parameter
  - If positive, no change
  - If negative, we return -value
    - If -5, then --5 = +5
- Similar
  - return
  - parameter

Function: Maximum of 2 numbers

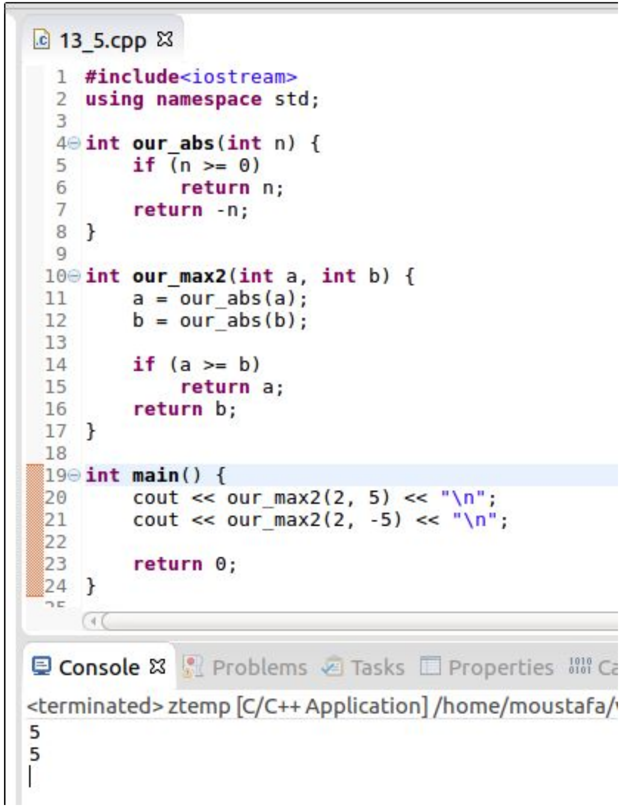


```cpp
#include<iostream>
using namespace std;

int our_max(int a, int b) {
    if (a >= b)
        return a;
    return b;
}

int main() {
    cout << our_max(2, 5) << "\n";
    cout << our_max(2, -5) << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/mousta
5
2
```

- Now we need to send the function 2 numbers not just 1
- We can send as much as we want

Function: Maximum of absolute 2 numbers

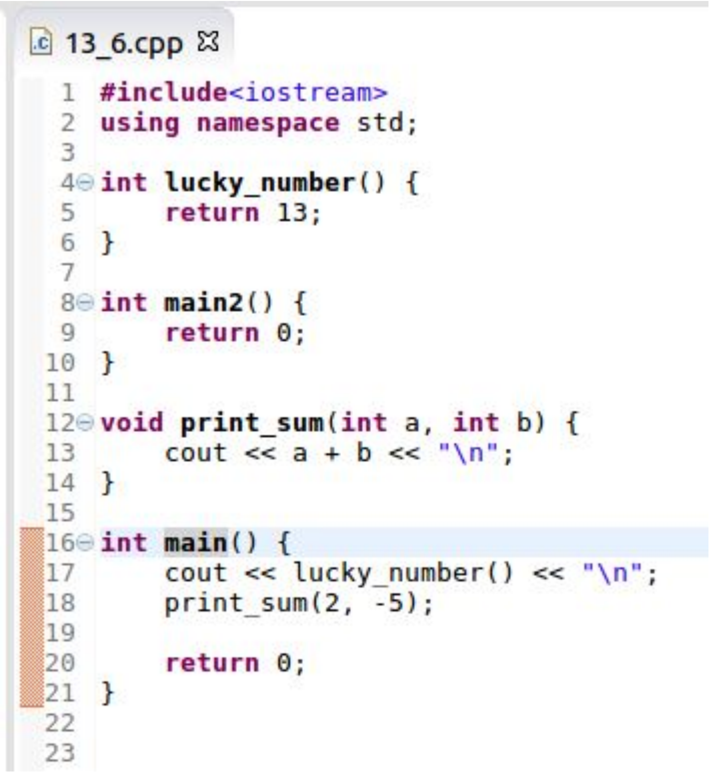


```cpp
#include<iostream>
using namespace std;

int our_abs(int n) {
	if (n >= 0)
		return n;
	return -n;
}

int our_max2(int a, int b) {
	a = our_abs(a);
	b = our_abs(b);

	if (a >= b)
		return a;
	return b;
}

int main() {
	cout << our_max2(2, 5) << "\n";
	cout << our_max2(2, -5) << "\n";

	return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/
5
5
```

- This time, the maximum should be applied on the absolute itself
- But we coded absolute!
- Let’s use it
- A function can call another function
  - Here max call abs function
  - It **must** be above it

Function with no parameter or no return

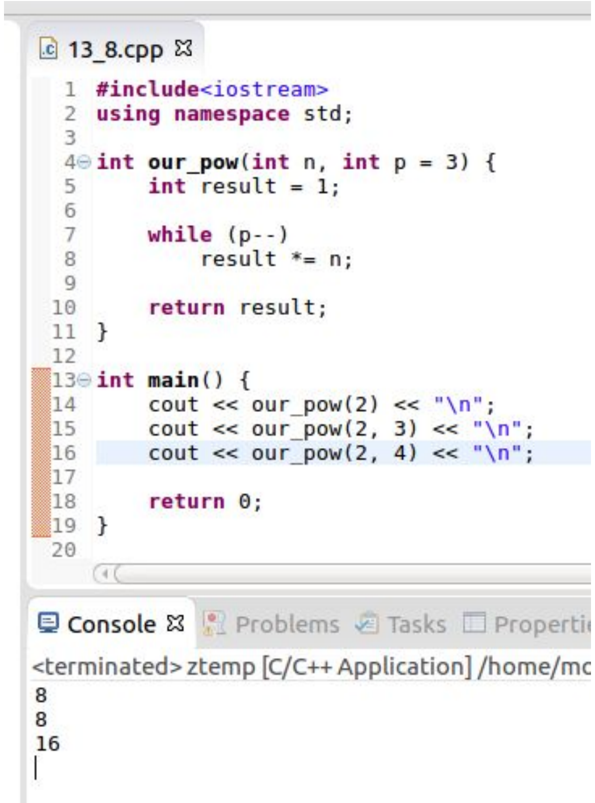


13_6.cpp

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

int lucky_number() {
    return 13;
}

int main2() {
    return 0;
}

void print_sum(int a, int b) {
    cout << a + b << "\n";
}

int main() {
    cout << lucky_number() << "\n";
    print_sum(2, -5);

    return 0;
}
```

- Sometimes, we don’t need to send parameters
- Sometimes we don’t return something
  - We use void
  - Common mistake: expecting return
- So what was main() ?

Function with default values

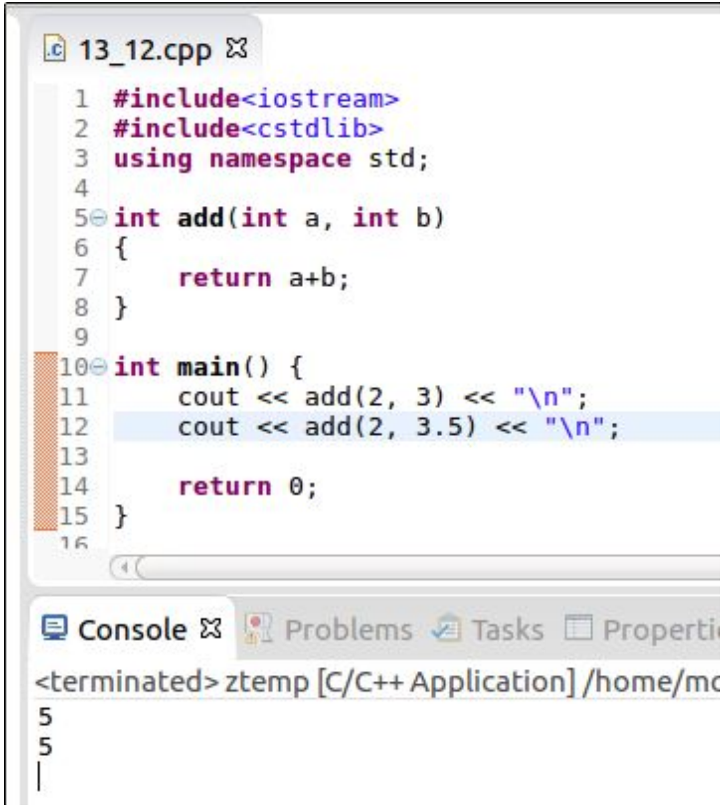


```cpp
#include<iostream>
using namespace std;

int our_pow(int n, int p = 3) {
    int result = 1;

    while (p--)
        result *= n;

    return result;
}

int main() {
    cout << our_pow(2) << "\n";
    cout << our_pow(2, 3) << "\n";
    cout << our_pow(2, 4) << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/mc
8
8
16
```

- Pow (2, 4) = 2 * 2 * 2 * 2
- P has default value 3
  - Means if u did not send it, 3 will be used
- Mistake
  - Can’t use variable without default value after a default value
  - Int solve(int a, int b = 2, int c); wrong
  - Int solve(int a, int b, int c = 3); correct

Be careful: Parameter casting

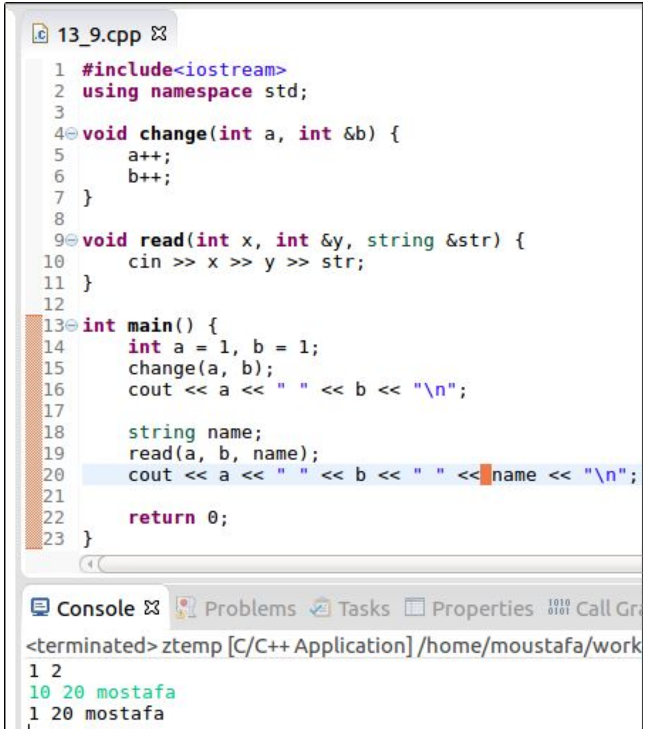


```cpp
#include<iostream>
#include<cstdlib>
using namespace std;

int add(int a, int b)
{
    return a+b;
}

int main() {
    cout << add(2, 3) << "\n";
    cout << add(2, 3.5) << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/mc
5
5
```

- What happens is:
  - Int b = 3.5
    - Which is 3 as integer
- When using function, notice carefully the casting

Pass variables: value vs reference

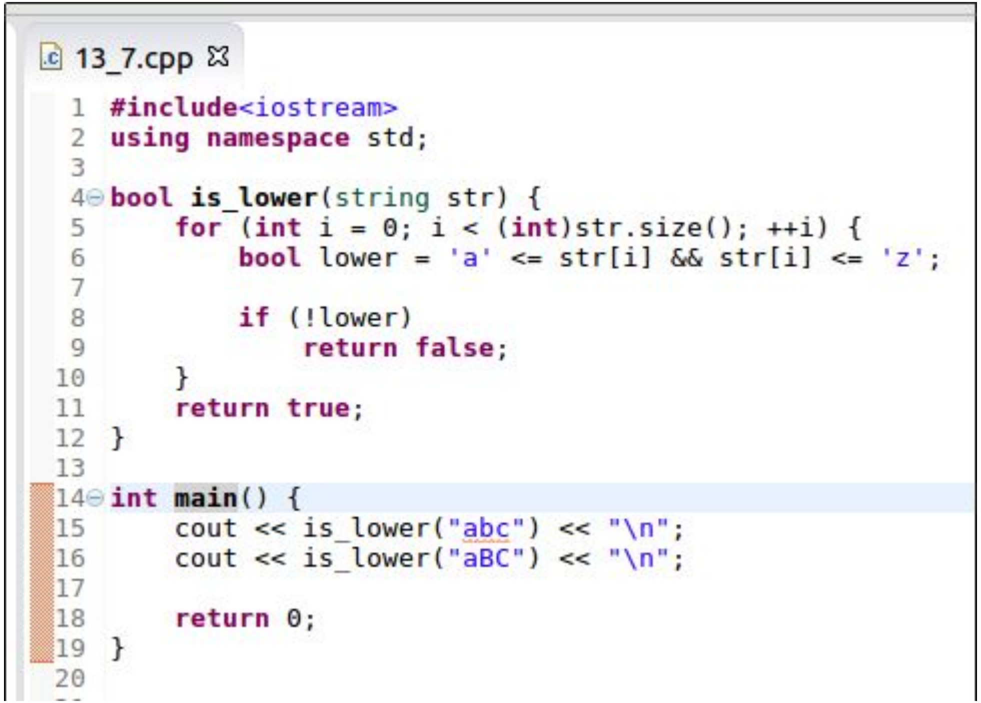


```cpp
#include<iostream>
using namespace std;

void change(int a, int &b) {
    a++;
    b++;
}

void read(int x, int &y, string &str) {
    cin >> x >> y >> str;
}

int main() {
    int a = 1, b = 1;
    change(a, b);
    cout << a << " " << b << "\n";

    string name;
    read(a, b, name);
    cout << a << " " << b << " " << name << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/work
1 2
10 20 mostafa
1 20 mostafa
```

- If you want the function change memory content of your variable, use &
- Otherwise it is copied

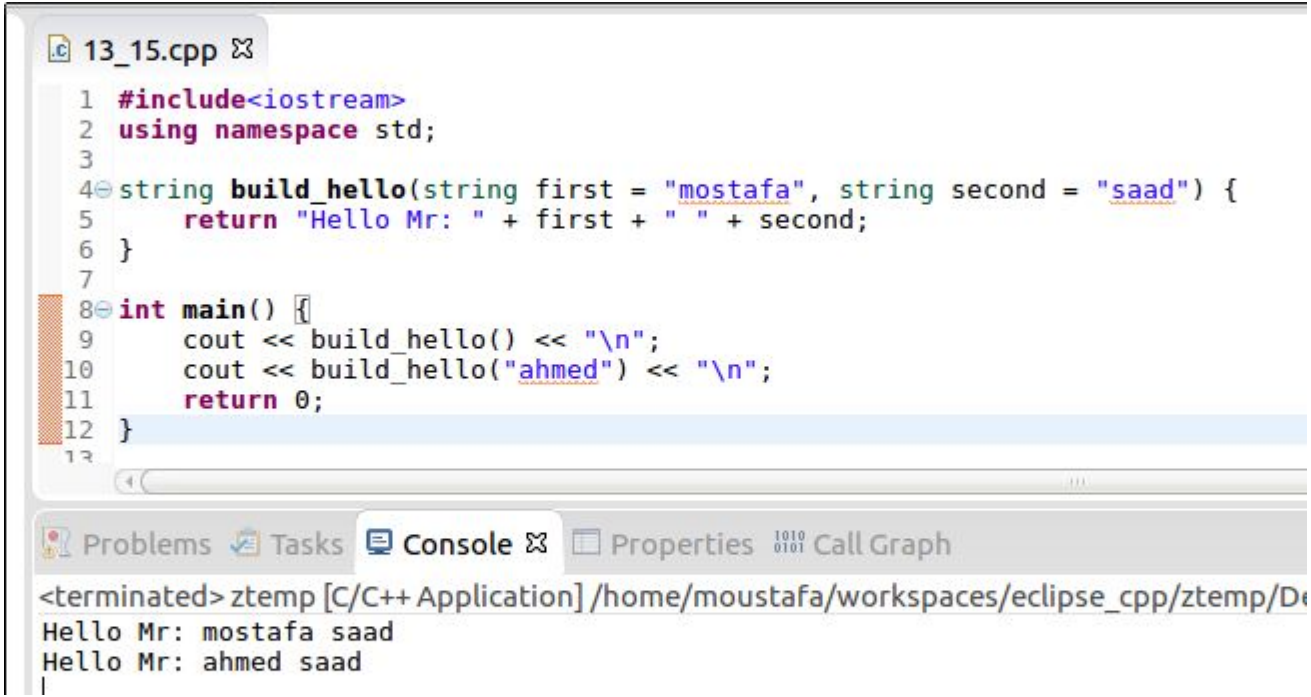
Function: Is lower string?



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

bool is_lower(string str) {
    for (int i = 0; i < (int)str.size(); ++i) {
        bool lower = 'a' <= str[i] && str[i] <= 'z';

        if (!lower)
            return false;
    }
    return true;
}

int main() {
    cout << is_lower("abc") << "\n";
    cout << is_lower("aBC") << "\n";

    return 0;
}
```

- We can also send strings!

Function: Concatenate strings

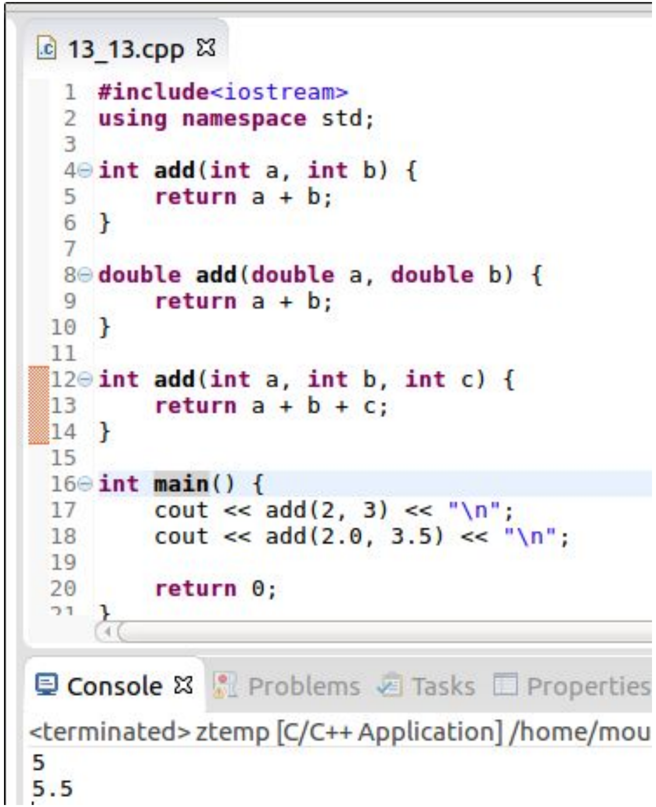


```cpp
#include<iostream>
using namespace std;

string build_hello(string first = "mostafa", string second = "saad") {
	return "Hello Mr: " + first + " " + second;
}

int main() {
	cout << build_hello() << "\n";
	cout << build_hello("ahmed") << "\n";
	return 0;
}
```

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

```
Hello Mr: mostafa saad
Hello Mr: ahmed saad
```

Function overloading

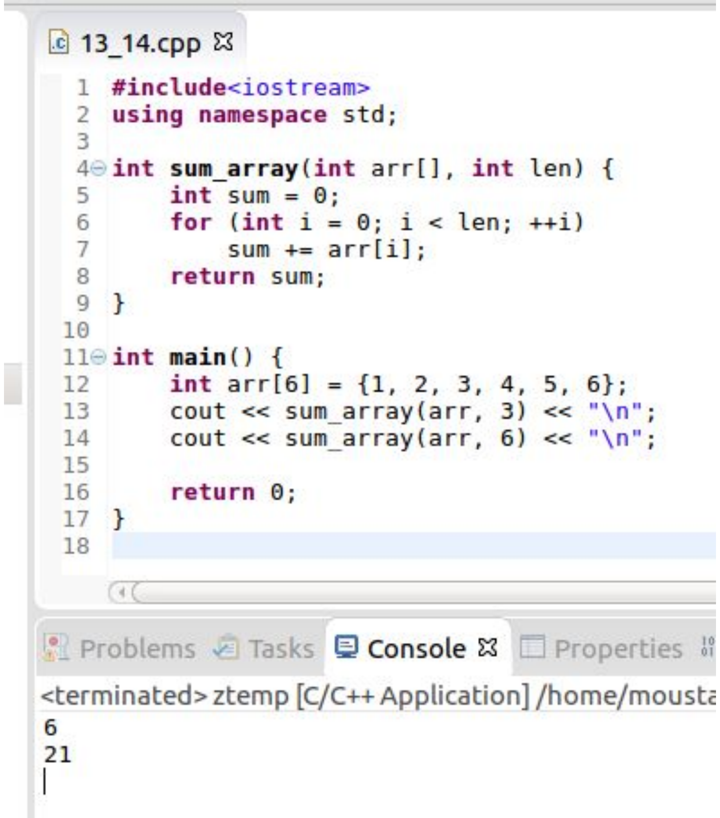


```cpp
#include<iostream>
using namespace std;

int add(int a, int b) {
    return a + b;
}

double add(double a, double b) {
    return a + b;
}

int add(int a, int b, int c) {
    return a + b + c;
}

int main() {
    cout << add(2, 3) << "\n";
    cout << add(2.0, 3.5) << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/mou
5
5.5
```

- We can define several functions with SAME name BUT different parameters (types or their count)

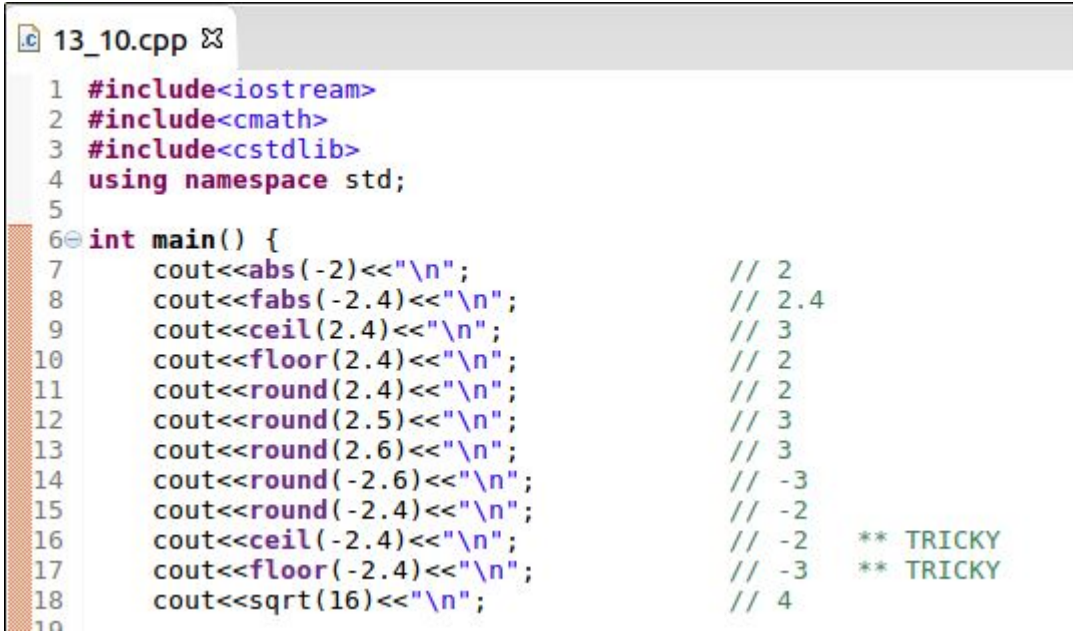
Function: Array as parameter (by reference)



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

int sum_array(int arr[], int len) {
    int sum = 0;
    for (int i = 0; i < len; ++i)
        sum += arr[i];
    return sum;
}

int main() {
    int arr[6] = {1, 2, 3, 4, 5, 6};
    cout << sum_array(arr, 3) << "\n";
    cout << sum_array(arr, 6) << "\n";

    return 0;
}

```

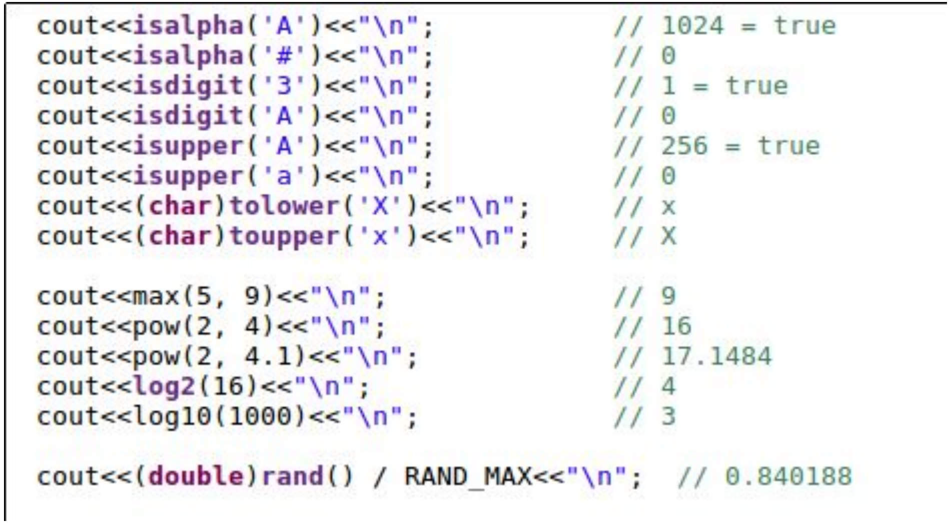
```
<terminated> ztemp [C/C++ Application] /home/mousta
6
21
```

Builtin functions



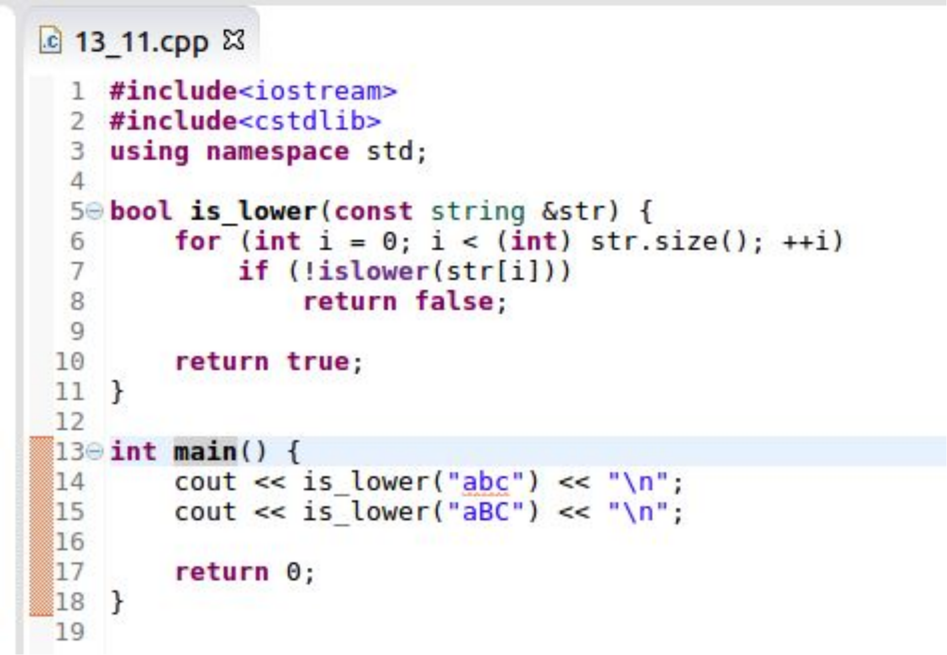
```cpp
#include<iostream>
#include<cmath>
#include<cstdlib>
using namespace std;

int main() {
    cout<<abs(-2)<<"\n";                // 2
    cout<<fabs(-2.4)<<"\n";             // 2.4
    cout<<ceil(2.4)<<"\n";              // 3
    cout<<floor(2.4)<<"\n";             // 2
    cout<<round(2.4)<<"\n";             // 2
    cout<<round(2.5)<<"\n";             // 3
    cout<<round(2.6)<<"\n";             // 3
    cout<<round(-2.6)<<"\n";            // -3
    cout<<round(-2.4)<<"\n";            // -2
    cout<<ceil(-2.4)<<"\n";             // -2   ** TRICKY
    cout<<floor(-2.4)<<"\n";            // -3   ** TRICKY
    cout<<sqrt(16)<<"\n";               // 4
```

Builtin functions



```
cout<<isalpha('A')<<"\n";           // 1024 = true
cout<<isalpha('#')<<"\n";           // 0
cout<<isdigit('3')<<"\n";           // 1 = true
cout<<isdigit('A')<<"\n";           // 0
cout<<isupper('A')<<"\n";           // 256 = true
cout<<isupper('a')<<"\n";           // 0
cout<<(char)tolower('X')<<"\n";     // x
cout<<(char)toupper('x')<<"\n";     // X

cout<<max(5, 9)<<"\n";              // 9
cout<<pow(2, 4)<<"\n";              // 16
cout<<pow(2, 4.1)<<"\n";            // 17.1484
cout<<log2(16)<<"\n";               // 4
cout<<log10(1000)<<"\n";            // 3

cout<<(double)rand() / RAND_MAX<<"\n";  // 0.840188
```

Function: Is lower string? again!



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

bool is_lower(const string &str) {
    for (int i = 0; i < (int) str.size(); ++i)
        if (!islower(str[i]))
            return false;

    return true;
}

int main() {
    cout << is_lower("abc") << "\n";
    cout << is_lower("aBC") << "\n";

    return 0;
}
```

Homework

- We can make a lot of mistakes when using functions
- Create as much mistakes as u can
    - E.g. mistake in function names, forget to return, call with wrong data types, call function doesn’t exist, etc
- Read the compilation error and understand it

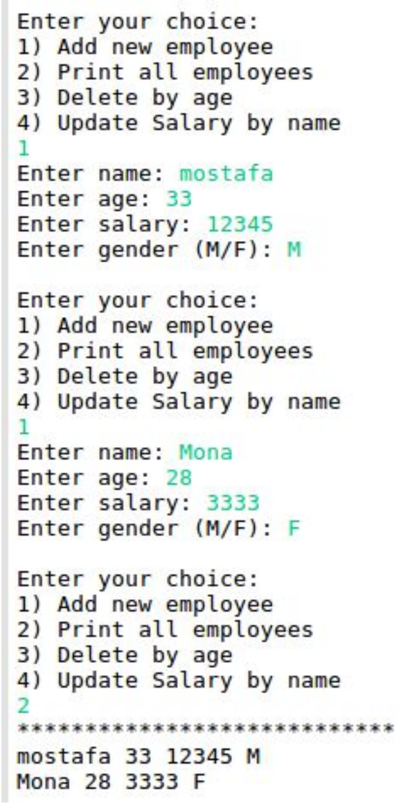
Practice: Employee Program (v2)

- Help our factory in managing his employees. Create a program that does the following:
    - Display the following choices:
        - Enter your choice:
        - 1) Add new employee
        - 2) Print all employees
        - 3) Delete by age
        - 4) Update Salary by name
    - You will keep the program running forever. Display the choices and user input from 1 to 4

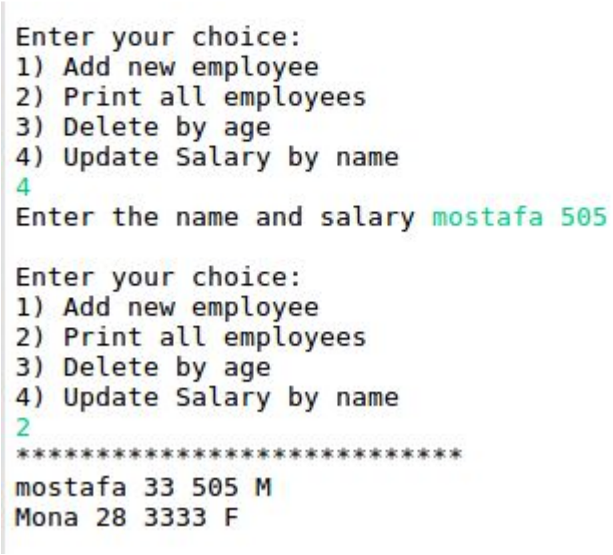
Practice: Employee Program (v2)

- For choice 1: Allow the manager to enter information of an employee
  - Ask user to input: Name, Age, Salary and Gender letter
  - Add the information to your database
- For choice 2: Print all employees. Line per employee
- For choice 3: User enter to values, start age and end age
  - Find all employees with: start_age <= age <= end_age and remove them
- For choice 4: User enter name, then salary
  - Find the employee and update his salary
- See screenshots

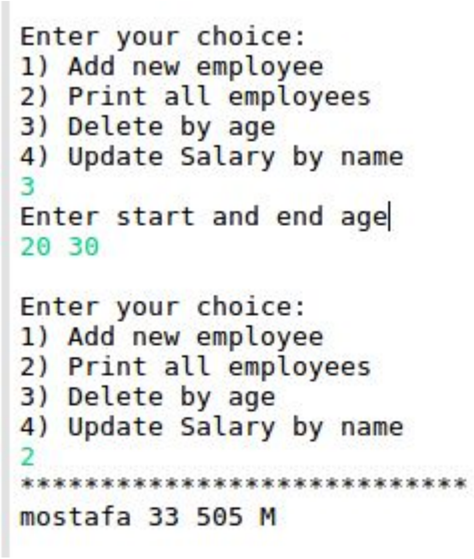
Practice: Employee Program (v2)



```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
1
Enter name: mostafa
Enter age: 33
Enter salary: 12345
Enter gender (M/F): M

Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
1
Enter name: Mona
Enter age: 28
Enter salary: 3333
Enter gender (M/F): F

Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
****************************
mostafa 33 12345 M
Mona 28 3333 F
```

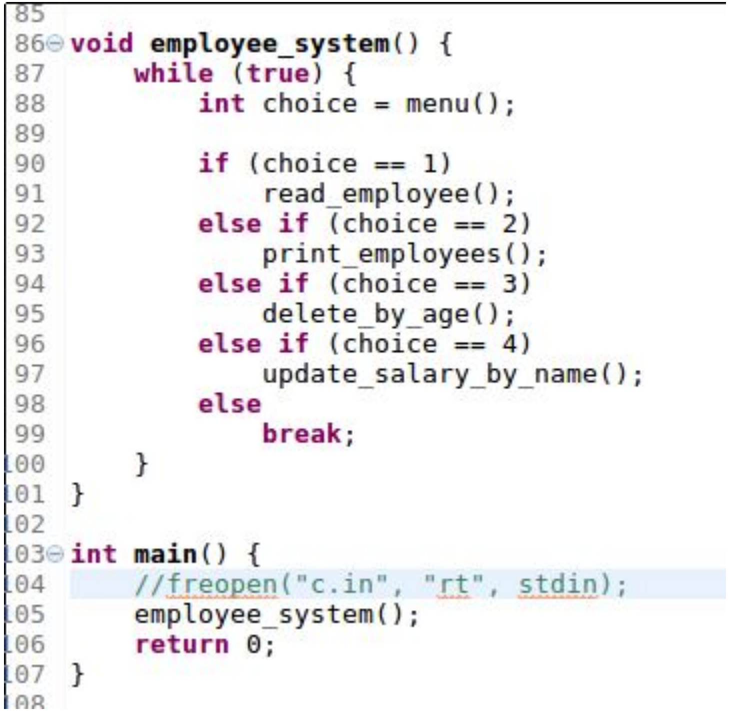


```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
4
Enter the name and salary mostafa 505

Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
****************************
mostafa 33 505 M
Mona 28 3333 F
```



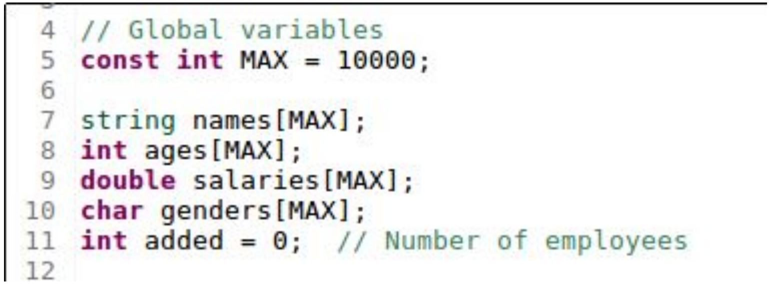
```
Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
3
Enter start and end age
20 30

Enter your choice:
1) Add new employee
2) Print all employees
3) Delete by age
4) Update Salary by name
2
****************************
mostafa 33 505 M
```

Practice: Employee Program (v2)



```
void employee_system() {
    while (true) {
        int choice = menu();

        if (choice == 1)
            read_employee();
        else if (choice == 2)
            print_employees();
        else if (choice == 3)
            delete_by_age();
        else if (choice == 4)
            update_salary_by_name();
        else
            break;
    }
}

int main() {
    //freopen("c.in", "rt", stdin);
    employee_system();
    return 0;
}
```

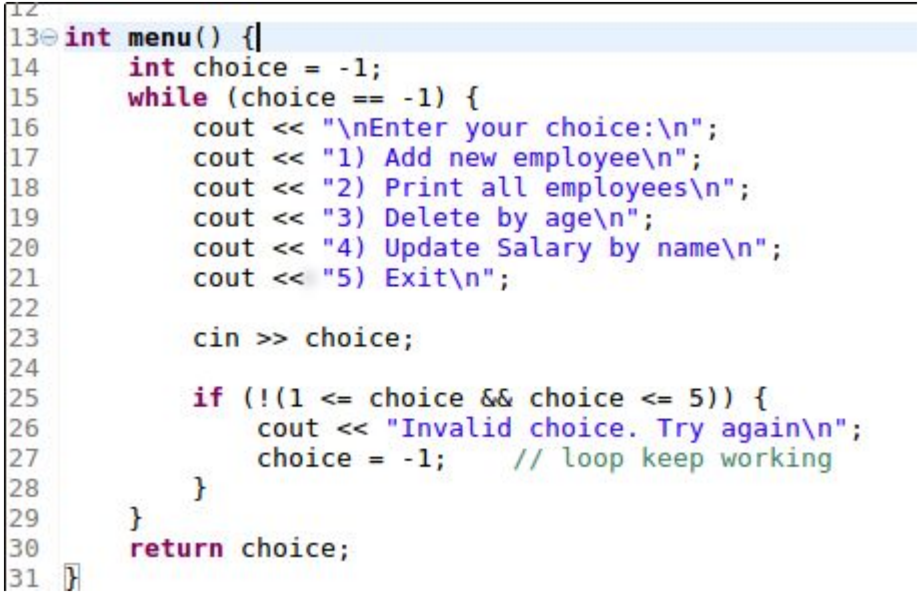
Practice: Employee Program (v2)



```cpp
// Global variables
const int MAX = 10000;

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

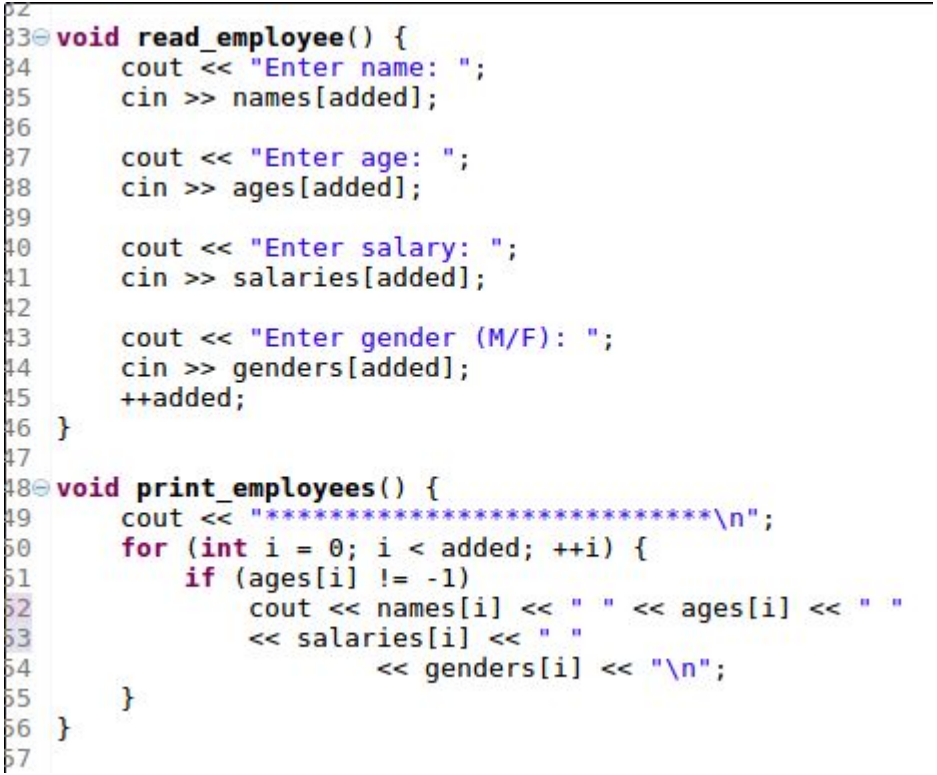
- Local variable
  - Defined inside function
  - Such as main or yours
- Global variable
  - Defined outside functions
  - Simpler, but not good professionally

Practice: Employee Program (v2)



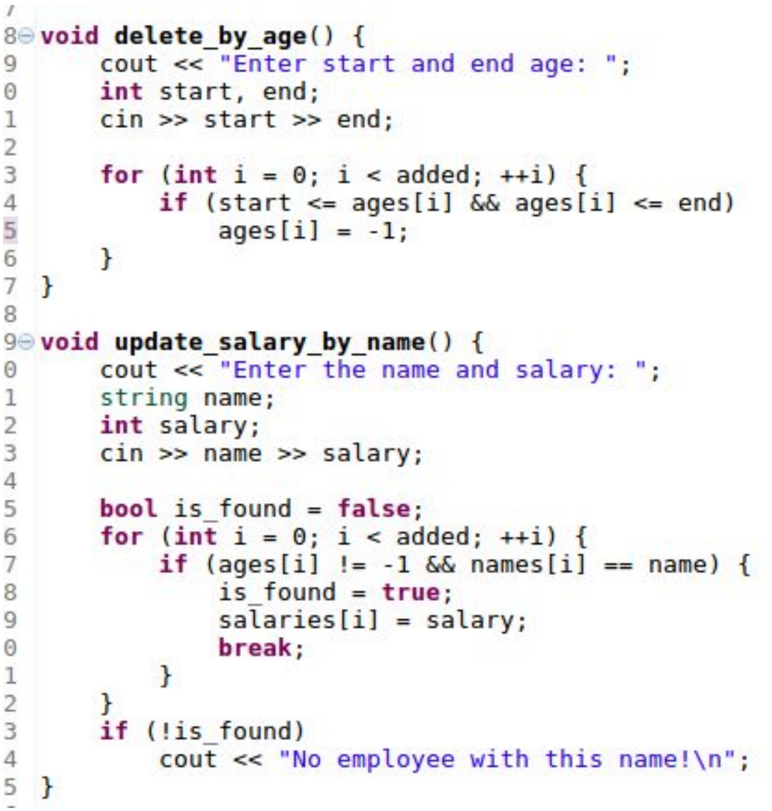
```
int menu() {
    int choice = -1;
    while (choice == -1) {
        cout << "\nEnter your choice:\n";
        cout << "1) Add new employee\n";
        cout << "2) Print all employees\n";
        cout << "3) Delete by age\n";
        cout << "4) Update Salary by name\n";
        cout << "5) Exit\n";

        cin >> choice;

        if (!(1 <= choice && choice <= 5)) {
            cout << "Invalid choice. Try again\n";
            choice = -1;    // loop keep working
        }
    }
    return choice;
}
```

Practice: Employee Program (v2)



```cpp
void read_employee() {
    cout << "Enter name: ";
    cin >> names[added];

    cout << "Enter age: ";
    cin >> ages[added];

    cout << "Enter salary: ";
    cin >> salaries[added];

    cout << "Enter gender (M/F): ";
    cin >> genders[added];
    ++added;
}

void print_employees() {
    cout << "****************************\n";
    for (int i = 0; i < added; ++i) {
        if (ages[i] != -1)
            cout << names[i] << " " << ages[i] << " "
            << salaries[i] << " "
                    << genders[i] << "\n";
    }
}

```

Practice: Employee Program (v2)



```cpp
void delete_by_age() {
    cout << "Enter start and end age: ";
    int start, end;
    cin >> start >> end;

    for (int i = 0; i < added; ++i) {
        if (start <= ages[i] && ages[i] <= end)
            ages[i] = -1;
    }
}

void update_salary_by_name() {
    cout << "Enter the name and salary: ";
    string name;
    int salary;
    cin >> name >> salary;

    bool is_found = false;
    for (int i = 0; i < added; ++i) {
        if (ages[i] != -1 && names[i] == name) {
            is_found = true;
            salaries[i] = salary;
            break;
        }
    }
    if (!is_found)
        cout << "No employee with this name!\n";
}
```

Homework 1: Max of 6 numbers

- Write a function that reads 6 numbers and compute their maximum. Create the following functions
  - max(int a, int b, int c)
  - max(int a, int b, int c, int d)
  - max(int a, int b, int c, int d, int e)
  - max(int a, int b, int c, int d, int e, int f)
- How can
  - max(int a, int b, int c, int d) utilize max(int a, int b, int c) ? and so on

Homework 2: Reverse a string

- Develop a function that do reverse for the string. Function is:
- string reverse_str(const string & str);
  - Don’t try to change str content or you will get compilation error

Homework 3: Calculator

- Develop a function that allows user to do the following (menu options):
    - Add 2 numbers
    - Subtract 2 numbers
    - Multiply 2 numbers
    - Divide 2 numbers
    - End the program
- Consider the following functions:
    - Function to read 2 double numbers - by reference
    - 4 functions, one for each operation. Don’t divide by zero!
    - Function to display the menu of the 5 options - read number and return it.
        - User should enter number from 1 to 5. If not, display error message
        - If exit, end the program by printing how many operations were done

Homework 4: Is Palindrome Array

- Read N, then N integers for an Array. Call a function with the array to check if the array is palindrome or note
  - We already coded it before
  - Just copy code and rearrange to call function with array

Homework 5: Set-powers

- Implement this function
- void set_powers(int arr[], int len = 5, int m = 2)
- This function will fill the array of len as following:
    - The i-th position: m^i, e.g. m * m * m … i times
    - E.g. for len = 6, m = 2 ⇒ 1 2 4 8 16 32
    - E.g. for len = 4, m = 3 ⇒ 1 3 9 27
- After a return from call: print the array
    - Try it with different default value scenarios

Homework 6: Get nth-prime

- Implement the following 2 functions:
- bool is_prime(int num);
    - Return true if number is prime
- Int nth_prime(int n);
    - Return the n-th prime number. It should use is_prime function
    - E.g nth_prime(6) = 13
        - Recall: 2, 3, 5, 7, 11, **13**, 17, 19

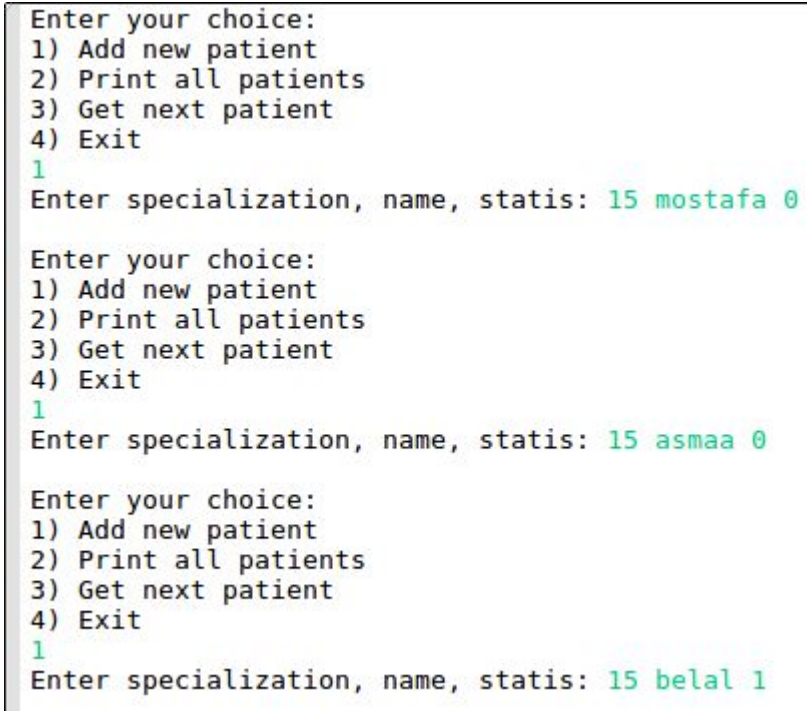
Homework 7: Replace substring

- Implement this function
- string **replace_str**(string input, string pattern, string to)
  - Constraints: Input consists only of lower cases, len(pattern) > 0, len(to) >= 0
- The function replaces every pattern with to and return it
  - Input: “aabcabaaad”, “aa”, “x” - Return: “**x**bcab**x**ad”
  - Input: “aabcabaaad”, “aa”, “aaaa” - Return: “**aaaa**bcab**aaaa**ad”
  - Input: “aabcabaaad”, “aa”, “” - Return: “bcabad”
- Let your code makes use of another function:
  - bool **starts_with**(string input, string pattern, int pos);
  - Return true if string input has the pattern starting from pos
  - Input: “aabcabaaad”, “aa”, 0 ⇒ True
  - Input: “aabcabaaad”, “aa”, 1 ⇒ False

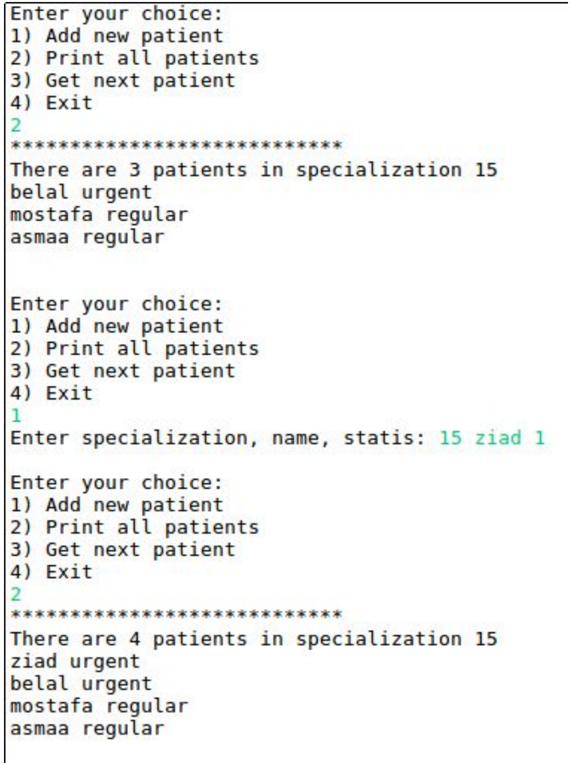
Homework 8: Hospital System

- Implement the following system for a hospital
- There are 20 different specialization (e.g. Children, Surgery, etc)
- For each specialization, there are only 5 available spots [queue]
- Adding a patient
  - Read the requested specialization [1-20].
  - Read his name and status (0 = regular, 1 urgent)
  - If 5 patients exist, apologize and don’t accept.
  - If the user is regular, add in end of queue. Otherwise, add in Begin
- Print patients, for the specializations that have waiting patients
- Dr pickup a patient
  - Read the requested specialization. If no patients, inform the doctor
  - Otherwise, ask the patient to go with the Dr. Remove from the queue

Homework 8: Hospital System

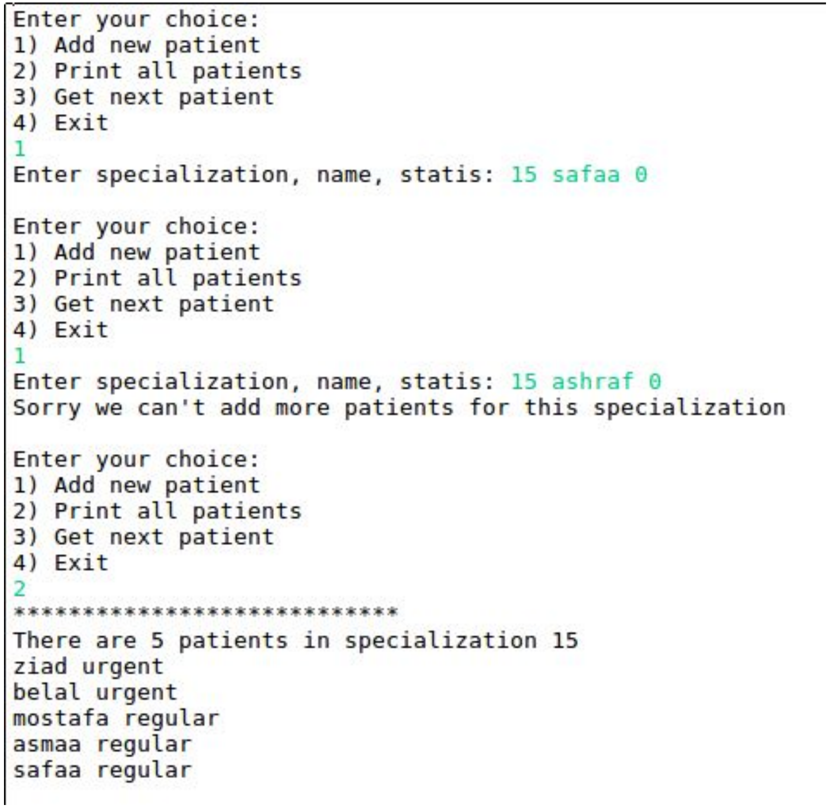


```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 mostafa 0

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 asmaa 0

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 belal 1
```

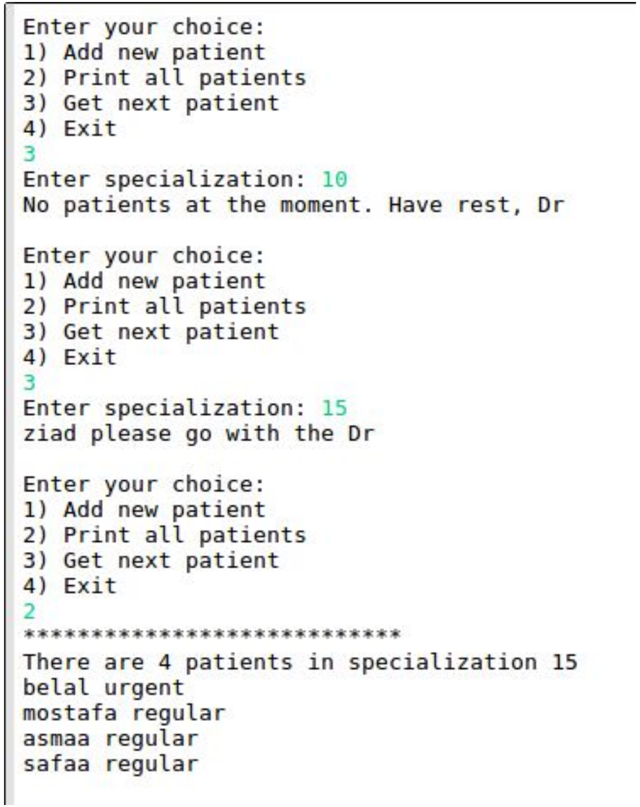


```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 3 patients in specialization 15
belal urgent
mostafa regular
asmaa regular


Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 ziad 1

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 4 patients in specialization 15
ziad urgent
belal urgent
mostafa regular
asmaa regular
```

Homework 8: Hospital System

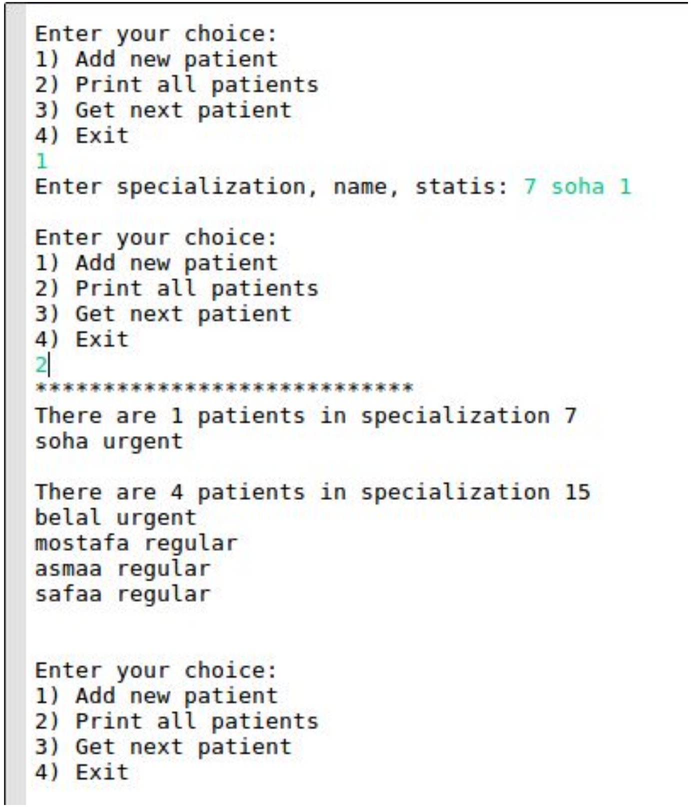


```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 safaa 0

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 ashraf 0
Sorry we can't add more patients for this specialization

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 5 patients in specialization 15
ziad urgent
belal urgent
mostafa regular
asmaa regular
safaa regular
```

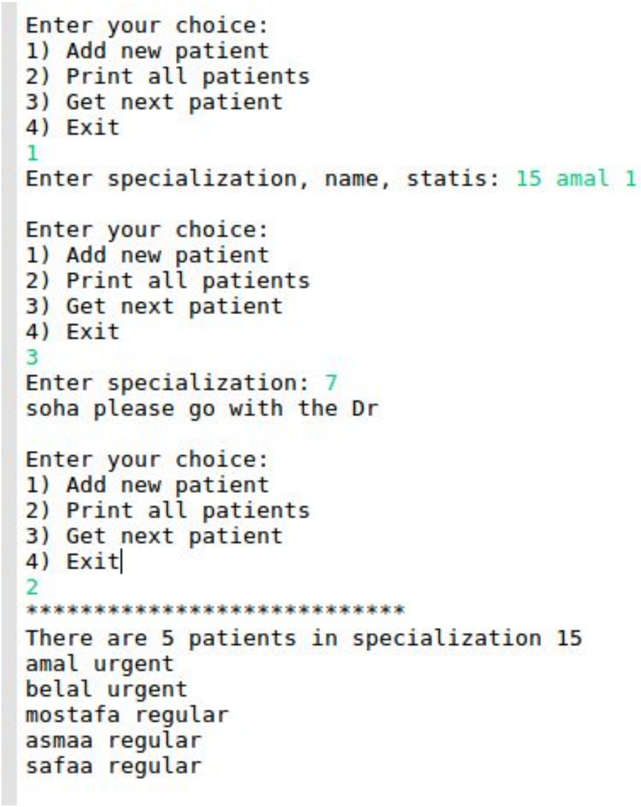


```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
3
Enter specialization: 10
No patients at the moment. Have rest, Dr

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
3
Enter specialization: 15
ziad please go with the Dr

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 4 patients in specialization 15
belal urgent
mostafa regular
asmaa regular
safaa regular
```

Homework 8: Hospital System



```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 7 soha 1

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 1 patients in specialization 7
soha urgent

There are 4 patients in specialization 15
belal urgent
mostafa regular
asmaa regular
safaa regular


Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
```



```
Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
1
Enter specialization, name, statis: 15 amal 1

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
3
Enter specialization: 7
soha please go with the Dr

Enter your choice:
1) Add new patient
2) Print all patients
3) Get next patient
4) Exit
2
****************************
There are 5 patients in specialization 15
amal urgent
belal urgent
mostafa regular
asmaa regular
safaa regular
```

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً