

# System and device programming

## Prof. Quer - 25 June 2024 - Standard Exam

	Meelad Dashti <b>s328715</b>
<b>Started on</b>	25 June 2024, 4:31 PM
<b>State</b>	Finished
<b>Completed on</b>	25 June 2024, 6:11 PM
<b>Time taken</b>	1 hour 40 mins
<b>Grade</b>	<b>7,75</b> out of 15,00 (52%)

### Question 1

Correct

Mark 1,25 out of 1,25

Suppose to run the following program. Indicate the possible admissible outputs. Note that wrong answers imply a penalty in the final score.

```
std::map<int, int> f (const std::vector<int>& vec, std::set<int> s) {
    std::map<int, int> m;
    for (const int &e : vec) {
        auto it = s.find(e);
        if (it != s.end())
            ++m[e];
    }
    return m;
}

int main() {
    std::vector<int> vec = {1, 2, 6, 2, 3, 3, 2, 3, 4, 5, 4, 4, 4, 5, 5, 5, 6, 6};
    std::set<int> s = {2, 4, 6};
    std::map<int, int> m = f(vec, s);
    for (const auto& pair : m) {
        std::cout << pair.first << ":" << pair.second << "-";
    }
    return 0;
}
```

- ☐ a. 1:1-2:3-3:3-4:4-5:4-6:3-
- ☐ b. 2:2-4:4-6:6-
- ☒ c. 2:3-4:4-6:3- ✓
- ☐ d. 1:1-2:3-3:3-4:4-5:4-6:3-
- ☒ e. 2:3-4:4-6:3- ✓
- ☐ f. 1:1-2:2-3:3-4:4-5:5-6:6-

Risposta corretta.

The correct answers are: 2:3-4:4-6:3-, 2:3-4:4-6:3-

### Question 2

Incorrect

Mark 0,00 out of 1,25

Analyze the following code snippet in C++. Indicate the possible output or outputs obtained by executing the program.

```
int main() {
    std::vector<int> vec = {1, 2, 3, 4, 5};
    auto lf = [&vec](long unsigned int index) -> int {
        if (index >= 0 && index < vec.size()) {
            int value = vec[index];
            return value;
        }
        std::cout << "Out" << " ";
        return -1;
    };
    std::cout << lf(2) << " ";
    std::cout << lf(5) << " ";
    std::cout << lf(0) << " ";
    return 0;
}
```

- ☐ a. 3 Out -1 1
- ☐ b. 3 -1 1
- ☐ c. 2 -1 0
- ☒ d. 3 Out 1 ✖
- ☐ e. 2 Out -1 0
- ☐ f. 3 Out -1 Out 1 Out

Risposta errata.

The correct answer is: 3 Out -1 1

### Question 3

Complete

Mark 2,25 out of 4,00

A C (or C++) program executes five threads: TA, TB, TC, TD, and TE. These threads are cyclical, run forever, and cooperate to generate sets of symbols on subsequent lines of the standard output. Each one can display one character, 'A', 'B', 'C', 'D', 'E', respectively, and a symbol "-" or/and a "new line" for each iteration of their primary cycle. Each line must have the following format:

A-{B|C}-{D&E}-A

This means that for each sequence, there are four sub-sequences separated by "-" symbols and terminated by a new line; the subsequences are the following:

- A: Exactly one symbol A.
- {B|C}: One symbol B or one symbol C.
- {D&E}: One symbol D and one E in any order (DE or ED).
- A: Exactly one symbol A.

The following are correct sequences generated by the program:

```
A-B-ED-A
A-C-ED-A
A-B-DE-A
A-C-DE-A
A-B-DE-A
...
```

```

#include <iostream>
#include <thread>
#include <counting_semaphore>

using namespace std;
mutex m;
counting_semaphore sA{1}, sB{0}, sC{0}, sD{0}, sE{0};
int count = 0;
bool BorC = false;

void TA(){
// as we want to print twice once in the beginning and once in the end. we put it in a for
for(i=0; i<2; i++){
    m.lock();
    sA.acquire(); //
    ++count;
    if(i==0) {cout<<"A" << "-";}
    if(count==1){
        sB.release(); // we release both the semaphores so that one of them runs
        sC.release();

```

Comment:

You do not need a "for" running twice for TA; you need "while(1)" loops for all threads. Threads must be cyclic not the main (threads must be created only once).

The m.lock on TB and TC is useless but just one must run, not bot.

The counters in TB, TC, TD, and TE are useless as well.

#### Question 4

Complete

Mark 1,50 out of 3,00

The following code implements the recursive merge sort procedure (without the merge function, which we do not have to care about).

```

void mergesort(std::vector<int>& arr, int left, int right) {
    if (left>=right)
        return;
    int mid = left + (right-left) / 2;
    mergesort (arr, left, mid);
    mergesort (arr, mid + 1, right);
    merge(arr, left, mid, right);
    return;
}

int main() {
    const int size = 12;
    std::vector<int> array;

    for (int i=0; i<size; i++)
        array.push_back(rand()%10000);
    mergesort(array, 0, array.size() - 1);
    std::cout << "Sorted array: ";
    for (int val: array) {
        std::cout << val << " ";
    }
    std::cout << std::endl;
    return 0;
}

```

Implement in C++ a parallel version using promises and futures. The main has to run a task, set the values into the array, and wake up the task to start the sorting process. Promises and futures can be used to return

the results of previously sorted subarrays or simply to synchronize tasks.

```
#include <iostream>
#include <future>
#include <vector>

using namespace std;

void mergesort(std::vector<int> &arr, int left, int right){
    if(left>=right)
        return;
    int mid = left + (right-left)/2;
    // for the recursive calls we use futures
    auto future = async(::launch::async, mergesort, arr, left, mid);
    auto future2 = async(::launch::async, mergesort, arr, mid+1, right);
    auto future3 = async(::launch::async, mergesort, arr, left, mid, right);
    future.get();
    future2.get();
    future3.get();
}

// I wasn't sure if this was required in the question because of ("run a task to set the
```

Comment:

last call in mergesort should be to merge

should be `promise<vector<int>>> &`

cannot use promise as argument of `set_value`

### Question 5

Complete

Not graded

By clicking on the following link, the **cheat sheet on C and C++ will open in a new windows**

Cheat Sheet C and C++

**If you want to withdraw** from Prof. Quer's exam, please **select true/vero/yes**. Otherwise, i.e., you want to take the exam, select false/falso/no. It is also possible to withdraw once the exam has been completed, sending an e-mail to the instructors.

- ☐ True
- ☒ False

The correct answer is 'False'.

### Question 6

Partially correct

Mark 0,75 out of 1,50

Analyze the following code snippet in C++. When the main is executed, Indicate which of the following statements are correct.

```

class C {
private:
    ...
public:
    ...
};
void f1(C &e) { cout << "{f1}";}
void f2(C *e) { cout << "{f2}";}
void f3(shared_ptr<C> e) { cout << "{f3}";}
int main() {
    cout << "{01}"; C e1;
    cout << "{02}"; C *e2 = new C[3];
    cout << "{03}"; shared_ptr<C> e3 = shared_ptr<C> (new C);
    cout << "{04}"; shared_ptr<C> e4 = shared_ptr<C> (new C);
    cout << "{05}"; f1 (e1);
    cout << "{06}"; f2 (e2);
    cout << "{07}"; f3 (e3);
    cout << "{08}"; e1 = (std::move(e2[0]));
    cout << "{09}"; delete[] e2;
    cout << "{10}"; return 0;
}

```

- ☐ a. Each one of lines number 3 and 4 includes one standard constructor and one copy assignment operator.
- ☐ b. Line number 7 includes one standard constructor.
- ☒ c. Line number 8 includes one move assignment operator. ✓
- ☐ d. Line number 9 includes three destructors.
- ☐ e. Line number 10 includes three destructors.
- ☐ f. Line number 1 includes one copy constructor.
- ☒ g. Line number 2 includes three standard constructors. ✓
- ☐ h. Line number 6 includes one standard constructor.
- ☐ i. Line number 5 includes one standard constructor.

Risposta parzialmente esatta.

You have correctly selected 2.

The correct answers are: Line number 2 includes three standard constructors., Line number 8 includes one move assignment operator., Line number 9 includes three destructors., Line number 10 includes three destructors.

### Question 7

Complete

Mark 1,85 out of 2,00

Explain what a condition variable in C (or C++) is and how it is typically used with mutexes. Describe a typical use case scenario for condition variables in a producer-consumer problem and why each construct must be inserted in the overall scenario.

Condition variables are used when we want a thread to wait until a certain condition has !  
Usually the thread waits until another thread sends a signal, after which the condition i  
We can understand this better in the following example.

```
pthread_mutex_t mutex;  
pthread_cond_t cv;  
int done = 0;  
  
void* T1(void *arg){  
    // here we lock the mutex so that when we are changing the value of d, and more important  
    // signal by accessing cv we don't face any conflict.  
    pthread_mutex_lock(&mutex);  
    done=1; // we set the value of the condition that has to be satisfied in order for main  
    // here and send the signal to the main thread so that it wakes up.  
    pthread_cond_signal(&cv);  
    pthread_mutex_unlock(&mutex); // after sending the signal we unlock the mutex so the o  
    // can access it.  
}  
  
int main(){
```

Comment:  
init syntax wrong

### Question 8

Complete

Mark 0,15 out  
of 2,00

Specify what are templates in C++ and why they are helpful.

After that, consider the following code snippet in which a class template is instantiated to store objects of different types. Define a class template that can be compiled and work with the main program reported below.

```
int main() {  
    Box<int> intBox(123);  
    Box<std::string> strBox("Hello");  
    Box<float> fB;  
    fB = seteValue (13.24);  
    std::cout << intBox.getValue() << std::endl;  
    std::cout << strBox.getValue() << std::endl;  
    return 0;  
}
```

```
template<typename:T>
class BOx{
    private:
    T t;
    Box(T t){
this.t=t}

// Ran out of timeee
can y
```

Comment:  
no description

incomplete code