1 Parameters and variables

(a) When a function is declared, you can choose between call by value and call by reference for the function parameters. Explain these two concepts clearly in your own words.

(b) When variables are declared, you can choose to make them local and global. Explain these two concepts clearly in your own words.

(c) There is a subtle difference between formal parameters and arguments. Explain these two concepts clearly in your own words.

Look at the following code, which defines two functions:

```cpp
#include <iostream>

using namespace std;

int range (int j, int limit)
{
    j = j + limit; limit = j - limit; j = limit - j;
    cout << "Range: \"" << j << "\" and \"" << limit << "\" endl;
    return (limit - j);
}

int sample (int i, int limit)
{
    int sum = limit;
    while ( i <= limit ) { sum += range (i,limit); i++; } //while
    cout << "Sample: \"" << i << "\" and \"" << limit << "\" endl;
    return sum;
}
```

These two functions can be called from a main, which makes use of global variables:

(d) The functions `range` and `sample` will be used with the following main:

```cpp
int x = 0;
int y = 1;
int top = 3;

int main()
{
    x = sample (y, top) - 3;
    cout << "The answer: \"" << x << "\" and \"" << y << "\" endl;
}
```

What will be the output of this program (not only the last sentence)? Explain shortly why.

(e) The function declarations are now changed to call-by-reference (int range (int &j, int &limit) and sample (int &i, int &limit)). What will be the output of this program? Explain shortly why.
(f) Instead of \( x = \text{sample}(y, \text{top}) - 3; \), now the call \( x = \text{sample}(x, x) - 3; \) is made in the \text{main}().

What will be the output be of these programs (for both the call-by-value and call-by-reference variants of the range and limit-functions. Explain shortly why for both variants.

2 Arrays

Consider the following Boolean array \( \text{map} \) with size \( m \) by \( n \). In the example below the size is \( m = 5 \) and \( n = 7 \). The character 'T' stands for the Boolean value \text{true}, the character 'F' for \text{false}.

\[
\begin{array}{cccccccc}
F & T & T & F & F & T & F \\
T & F & F & T & F & T & F \\
T & T & F & F & T & F & F \\
F & T & T & F & T & F & F \\
T & F & T & T & F & F & F \\
\end{array}
\]

(a) Design a C++-function \( \text{inbetween}(\text{map}, \text{row}) \) that counts how often the character 'F' is inbetween two 'T'-values, for the row \( \text{row} \). For instance in the example, for the second row the answer should be 1.

(b) Design a Boolean C++-function \( \text{invisible}(\text{map}, \text{col}) \) that returns true if a column \( \text{col} \) consists of only the character 'F'. For instance in the example, this is only true for the last column.

(c) The given map can be interpreted as the result of a pathplanner. The values on the map should be interpreted as follows; in case of a value 'T' one should go the right, in case of a value 'F' one should go down. Design a C++-function \( \text{longest\_path}(\text{map}, \text{r}, \text{c}) \) which indicates the start location \( \text{r}, \text{c} \) of the longest path possible on a given \( \text{map} \). The function should return the length of the found path. For instance in the example, this path which starts at (0, 1) has a length 8. When multiple paths have the same length, it is allowed to return one of the startlocations (your choice).

3 Structures and classes

(a) How many \text{public:} sections are required in a class for the class to be useful?

(b) How many \text{private:} sections are required in a class.

(c) When you define a C++ class, should you make the member variables public or private? Should you make the member functions public or private?

(d) When you define a C++ class, which items are considered part of the interface? What items are considered part of the implementation?

(e) When does a class have a default constructor automatically?

4 Inheritance

Consider the inheritance of the following classes:

\begin{verbatim}
class LaserScan
{
public:
    LaserScan();
    LaserScan(int nn);
    void MaxRange();
    void NearestObject();
private:
    int n;
};
\end{verbatim}
class DepthMap: public LaserScan
{
public:
    DepthMap(int nn, double dd);
    void Skeleton();
private:
    double d;
};

(a) How many public members does an object of class DepthMap have?
(b) Will the following code work? Explain why.

int main()
{
    DepthMap d(180,180);
    cout << d.n;
    return 0;
}