Innopolis University
Summer Bootcamp 2025
Computer Science and Engineering Intro Course
Final Test
23d August 2025

ANSWERS

Student Name:

Student Email:

Student Telegram:

Academic Group:

Exam Room:

(The form below is filled up by instructors)

Grade Summary	
Question	Grade (up to 1 point each)
Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	
Total (max. 10)	

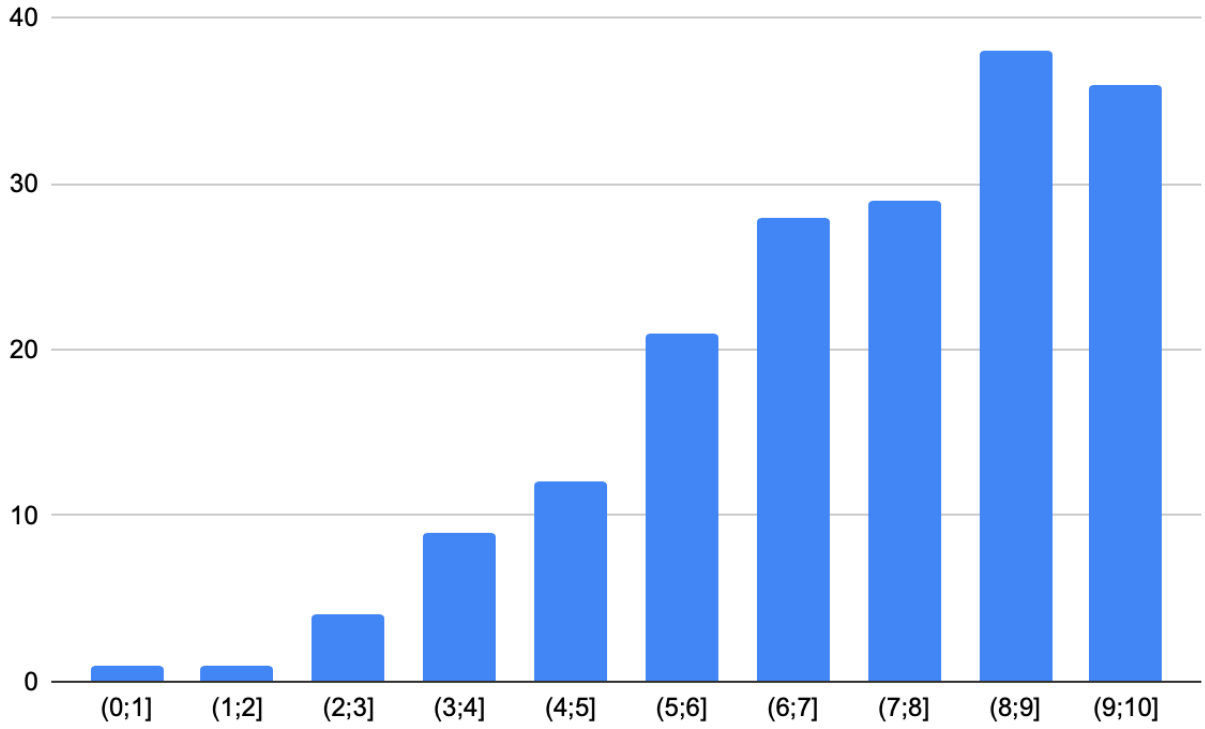
Please write very clearly your names and your solutions.
What will not be understood will be considered wrong.

You need to perform this exam alone and without the use of any equipment or book, apart from this booklet and pens. Please do not consult any person or use any equipment, otherwise you will be disqualified from this exam at the very first attempt.

Good Luck!

Your signature:

The resulted distribution of course grades:



Question 1 (1 point). Do you agree that, unlike C programming language, programs in Python require to be compiled before execution?

a) Agree
b) Disagree (correct, 1 point)

Question 2 (1 point). Consider the $n!$ factorial function implementation in C below.

```
unsigned long func(unsigned n) {
    if (n == 1) return 1;
    else return func(n-1)*n;
}
```

Choose its type (one option only):

a) Object-oriented;
b) Tail-recursive;
c) Non-tail-recursive (correct, 1 point);
d) Iterative

Question 3 (1 point). Order the following programming languages in a chronological order of their appearance: Go, Fortran, C, Python, C++

Answer:

1 (oldest)	**Fortran**
2	**C**
3	**C++**
4	**Python**

5 (youngest)	**Go**

Grading note:

1 – the entire order is correct;
0.5 – just two languages at neighbor positions are misplaced;
0 - otherwise

Question 4 (1 point). Very briefly describe any two advantages of C over Python:

Some potential advantages (among others):

1:	A higher memory efficiency
2:	A higher runtime efficiency

Grading note:

0.5 – for each correct option

Question 5 (1 point). What is the console output of the following C program?

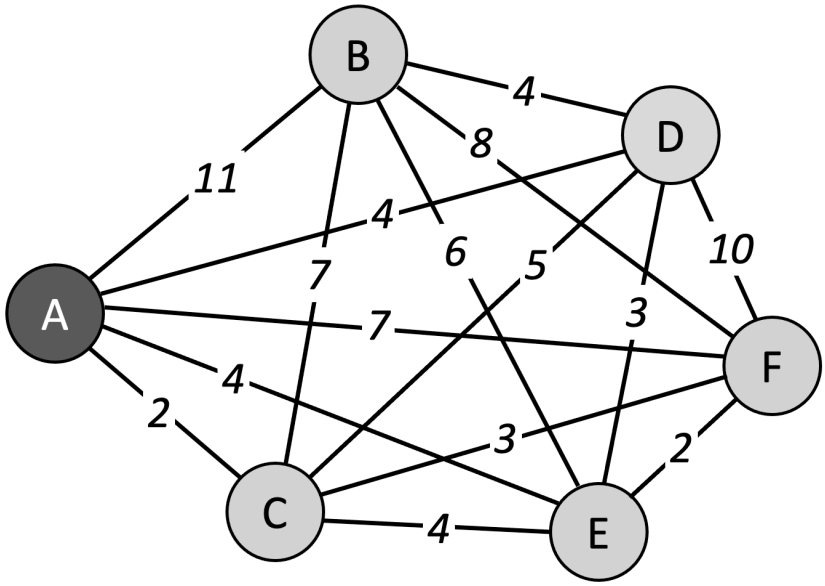
```c
#include <stdio.h>
int main() {
    int x = 1;
    int y = 2;
    int z = 3;

    for (int i = 2; i > 0; i--){
        z = (x + z) % y;
    }
    printf("%d", z);

    return 0;
}
```

Answer: **1**

Question 6 (1 point). Solve the TSP problem for the input graph below by using **the nearest neighbor heuristic/approach**.
The route starts and ends at city "A", as well as every city must be visited once and only once. The aim is to find a suboptimal short route.



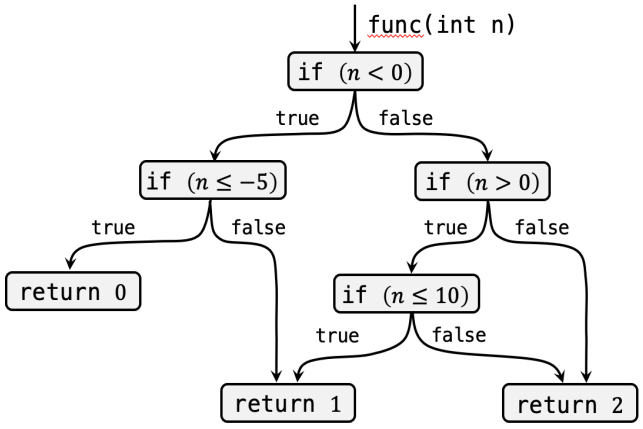
Suboptimal route: **Its length:** **25**

A	C	F	E	D	B	A

Grading note:

1 – a nearest neighbor suboptimal route and its length are both specified correctly;
0.5 – a suboptimal route is correct, but its specified length is incorrect;
0 – otherwise (including the case when an optimal path is specified instead of the suboptimal)

Question 7 (1 point). For a given execution tree below, write the corresponding function code for `func(int n)` in C programming language.



Answer:

```c
int func(int n) {
    if (n < 0) {
        if (n <= -5) {
            return 0;
        } else {
            return 1;
        }
    } else {
        if (n > 0) {
            if (n <= 10) {
                return 1;
            } else {
                return 2;
            }
        }
        return 2;
    }
}
```

Grading note:
1 – for a correct program;

0 – otherwise.
A few minor syntax mistakes (such as ";" forgotten) are ignored

Question 8 (1 point). Consider a bucket of 5 different balls. How many possible ways exist to pick up 3 of these balls, if the order in which balls are picked up matters?

Answer (computations are not mandatory): **60 (1 point)**

Computations (not graded): based on the lecture slides

Question 9 (1 point). Consider a pair of fair dice with 6 faces each. Let us roll this pair of dice for 2 times. What is the probability that at each roll the sum of dice faces equals "5"?

Answer (computations are not mandatory): **1/81** or **0.01234… (1 point)**

Computations (not graded): based on the lecture slides

Question 10 (1 point). Answer the following questions about the "`gcc`" program:

Some of the correct answers expected:

What is the purpose of the "`gcc`" program?	To compile C programs (into executable files)
What is the use of its "`-o`" option?	To specify the name of a resulted executable file
What is the use of its "`-lm`" flag?	To link the "math.h" library
What is the use of its "`-O3`" flag?	To apply various runtime optimizations on code during the compilation process

This is the end of the student book