

Please write on each paper the date and the name of the course as well as your name, student id and signature. Write also your mooc user name to the exam paper.

The exam has 4 questions and there will be 20 points available. Remember to write necessary comments to your functions and programs.

1) Function CelsiusToFahrenheit (2 points)

Write a C function `CelsiusToFahrenheit` that gets the Celsius value as float parameter and returns the corresponding Fahrenheit value. The conversion can be done using the formula $F = C * 9/5 + 32$.

2) Program convert (5 points)

Write the whole program including `main` function for a C program `convert`, that takes as command line arguments information about conversion and the value to be converted. At this point your program can only do conversions from CelsiusToFahrenheit using the function from question 1, but there are plans to add more conversions later. So check that the conversion indication is correct and prepare for the additions. The program will be called from command line:

```
convert -CtoF 25.6
```

Here the argument `-CtoF` indicates the conversion and the number (25.6) the value to be converted. Please give a simple help output to the user if the user did not provide the required command line arguments. Give the same help output when the program is started as `convert -help`.

(*HINT: argv, argc, double atof(const char *nptr)*).

3) Finding the longest line in a text file (6 points)

Write a C program to determine the line number of the longest line in the file. You may ask the user for the file name or get it as command line argument. The program only prints the line number, not the line itself. Please number the lines starting from line 1. If there are several lines with the same length, you only need to print line number of the first.

4) Linked list (7 points)

You have been given the following type definitions in a file `friends.h`. The given definitions are for a linked list where the `FriendListT` contains the number of friends in the list and pointer to the first element. Elements are reached using the next pointer in each element. Each element points to the next one in the list.

```
typedef struct {
    char name[100];
    FriendT *next;
} FriendT;

typedef struct {
    int length;
    FriendT *first;
} FriendListT;
```

Write two functions `AddFriend` and `RemoveFriend` to add and remove friends. These functions get as parameters the friend's name and pointer to the structure `FriendT`.

Explain how you assume the structure to be used, e.g. are the names in a sorted order or not.