University of Leeds **School of Computing**
Procedural Programming COMP1711
Semester 1, 2018-2019

Assignment 1

20 Marks
To be submitted before 5th November at 10am

The Brief

You will write a program for a small business (company). The program computes and prints the net pay for different employees at the end of each month. It also computes and prints a summary of the payroll for that month.

The Details

The Problem

A company requires a computer program to compute the pay and print a detailed payslip (also called pay advice) for each of its employees.

All the employees in the company are *salaried employees*. A salaried employee is one who has a fixed annual salary, and is paid the same amount every month.

Employees cannot take home all their monthly earnings. They have to pay two types of taxes: **income tax** and **national insurance contribution**. The amount of income tax and national insurance contribution depends on the person's annual salary. The annual salary is always quoted before any taxes are deducted.

For salaried employees, the net monthly pay, or *take-home money*, is calculated as follows:

1. The annual salary is divided by 12 to obtain the monthly pre-tax salary.
2. The monthly income tax and national insurance contribution are computed according to the procedures described below.
3. The net monthly salary is computed by deducting the income tax and national insurance contribution from the monthly pre-tax salary:

   Net Monthly Pay = Pre-Tax Monthly Salary – Monthly Income Tax – Monthly National Insurance Contribution

Calculating Income Tax

In the UK, the amount of income tax a person pays depends on annual income. The income tax rate (percentage) rises in steps as the annual income rises over certain discrete bands. Income at higher bands is charged a higher tax rate than lower bands, as in the following table:

Band No.	Band Name	Taxable Income	Tax rate
1	Basic rate	Up to 34,500	20%
2	Higher rate	£34,501 to £150,000	40%
3	Additional rate	over £150,000	45%

Fortunately, not all the money one earns is taxed. Each person is entitled to a certain amount of tax-free earning called the *personal allowance*. Before calculating income tax the personal allowance (i.e. the tax free portion of the income) is deducted from the annual salary. This personal allowance is fixed at £11,850 as long as the annual salary is less than or equal to £100,000. However, **if the annual salary is greater than £100,000, then the personal allowance is reduced by £1 for each £2 earned over £100,000**. Clearly, the personal allowance will reach 0 when the annual salary is £123,700 or higher (why?).

Hence, to compute the monthly income tax for an employee:

1. Calculate the adjusted personal allowance (if the annual salary is greater than £100,000)
2. Deduct the personal allowance from the annual salary to obtain the *taxable income*.
3. Calculate the annual income tax using the above table.
4. Divide the annual income tax by 12.

Examples:

1. An employee has an annual salary of £8500

The annual salary of this employee is less £100,000, hence, their personal allowance is fixed at £11,850. Also, since the annual income is less than the personal allowance, they pay 0% income tax on their salary (i.e. do not pay any income tax). The monthly income tax is:

Monthly income tax = (0% * 8500) /12 = £0.0

2. An employee has an annual salary of £25300

The annual salary of this employee is less £100,000, hence, their personal allowance is unchanged. The taxable income is 25300 – 11850 = 13450. This amount is less than 34,500 therefore they pay 20% on this amount:

Monthly income tax = (20% * 13450) /12 = £224.17

3. An employee has an annual salary of £96300

The annual salary of this employee is less than £100,000, hence, their personal allowance is unchanged. The taxable income is 96300 – 11850 = 84450. This amount has reached band 2, therefore they pay 20% on the first 34500 of their taxable income and 40% on the remaining amount:

Monthly income tax = **(** 20% * 34500 + 40% * (84450-34500) **)** /12 = £2240.00

4. An employee has an annual salary of £121,500

The annual salary of this employee is greater than £100,000, hence, their personal allowance is reduced by £1 for each £2 they earn over 100,000:

Personal allowance = 11,850 – 1 * **(** (121,500 - 100,000)/2 **)** = 1,100

The taxable income is 121,500 – 1,100 = 120,400

The taxable income has reached band 2, therefore they pay:

Monthly income tax = **(** 20% * 34500 + 40% * (120,400-34500) **)**/12 = £3438.33

Calculating National Insurance Contributions (NIC)

The amount of NIC depends on the person's age and annual salary. If the person is above the age of retirement (65 years), they pay 0% for national insurance, otherwise the national insurance contribution is computed as follows:

- If the person earns less than the National Insurance threshold which is currently set at £8,424 a year, they pay no National Insurance contributions.
- If they earn above the threshold, they pay 12% of their earnings between £8,424 and £46,350.
- On anything they earn above £46,350 a year, they pay National Insurance at 2%.

The Task

Write a C program that can be used by the above company to:

1) Compute and print the net monthly pay for any employee.
2) Compute the following monthly sums and statistics:
   A. The total amount of money paid out by the company to all employees.
   B. The average net monthly salary.
   C. The maximum income tax paid by any employee.
   D. The minimum income tax paid by any employee.

The program should be menu driven, i.e. when you run the program a menu similar to this one should appear:

```
1. Compute and Print the Pay for an Employee
2. Show Sums and Statistics
3. Quit the program

Please enter your choice (1, 2, or 3):
```

The program waits for the user to enter the number of an option. If the user enters 1, the program prompts the user to enter the annual salary of an employee and their age in years. The program then computes the monthly pay and **displays the details on the screen (an electronic payslip)**. The pay details should include the following:

- The age of the employee
- The annual salary
- The taxable income
- The monthly income tax
- The monthly NIC
- The net monthly pay

After 'printing' the pay advice, the program displays the menu again and waits for the user to enter a choice. If the user selects choice number 2, the program displays the sums and statistics mentioned above, and redisplays the menu. Finally to quit the program the user enters 3.

General Guidelines

- In this program you will be using the following C statements only:
  - Variable declaration statements.
  - The printf and scanf functions.
  - The assignment statement.
  - The conditional statements (e.g.'if else').
  - Iteration statements (loops).
- Please do **NOT** use arrays, structs, or pointers even if you are already familiar with them.
- Write the program in standard C. If you write your code in any other language, it will NOT be assessed and you will get a zero mark.
- This is an **individual** project, and you are not supposed to work in groups or pairs with other students.
- Be aware that plagiarism in your code will earn you a zero mark and will have very serious consequences. It is much better to submit your own partially finished work, than to fall into the trap of plagiarism.
- We use fairly sophisticated software which can identify whether two pieces of code are nearly identical (modifications such as renaming variables and reshuffling the positions of functions do not deceive it). If two (or more) students have large portions of their files nearly identical they will be accused of plagiarism or collusion. If found guilty, all parties involved will incur the penalty, regardless of who was the author of the code. For this reason, never show, or give access to, your code to anyone. Do not help a colleague by sharing your code, or you will both be found guilty of collusion.
- It is your responsibility to make sure that nobody has access to your code. Lock the session if you leave your computer unattended.

Submission

- Submit a single text file with extension *.c* through Minerva.
- Download and check the submitted file, to make sure that it is the correct version. We will not accept late submissions if you realise you uploaded the wrong file, or the file appears to be corrupted.
- Binary files (rather than text), and generally files that cannot be compiled, will earn zero marks.

Writing and testing your program

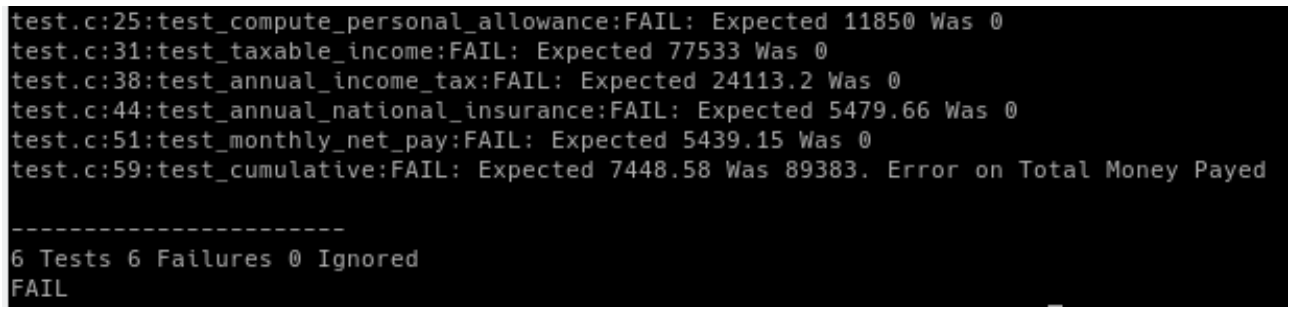
Part of the submission will be automatically evaluated by using the attached test harness. In order for the tests to work, your code must contain some predefined functions. You can find those functions along with their documentation, in the file *template.c*, which should therefore be your starting point. Rename this file, and implement the functions in it one by one. To run the tests, unpack the zip file containing the test harness in the same directory where your program is, and compile it with the following command:

```
gcc -lm -std=c99 –o tests unity.c test.c your_program.c
```

where *your_program.c* is the name of your program file. Each executable can have a single `main()` function, therefore, when compiling the tests, rename your main function into something else, for instance `mymain()`. If the test program compiles correctly, you can then execute it by running:

```
./tests
```

Initially, when all the functions are empty, all the tests will fail, and you will get an output like the following screenshot:



Once you have successfully implemented the body of the functions, all the tests will be passed.

Once you are satisfied with the tested functions, you can develop the menu interface and the main function. The functions that are automatically tested, that is the ones you initially found in the file *template.c*, **must not** contain any interactive component. This means that they must not use printf or scanf and must run without user input. Otherwise, the tests will not run automatically. The development of the interface must be separate from the development of the functionality of the program, as is good practice.

Support

Please take advantage of lab sessions to get support from the instructors, especially on being able to run the tests, since this is critical to passing the assignment.

Marking Scheme

The program passes the compute_personal_allowance test	(1 mark)
The program passes the taxable_income test	(1 mark)
The program passes the annual_income_tax test	(3 marks)
The program passes the annual_national_insurance test	(3 marks)
The program passes the monthly_net_pay test	(2 marks)
The program passes the comulative test	(2 marks)
The program displays the menu and processes user options correctly	(4 marks)
The program is well structured, clear, and efficient	(3 marks)
The program uses comments properly	(1 marks)