Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

# ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9

з навчальної дисципліни “Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУЖЕНИХ ТА ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ВИКОНАВ

студент академічної групи

КІ-21-1 Стрюк В. Є.

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_  П.С. Усік

Кропивницький – 2022

# ЛАБОРАТОРНА РОБОТА №9

**Тема:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів.

**Мета роботи:** полягає у набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів С/С++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів у кросплатформовому середовищі Code::Blocks.

## Варіант 6

**Завдання:**

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).
2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.

# ХІД РОБОТИ

### **Завдання 9.1**

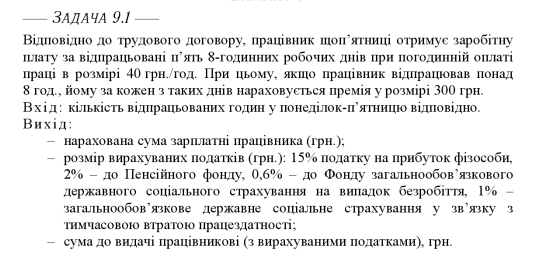


Рисунок 1.1 - Умова задачі 9.1

### **Строга постановка задачі:**

Вхідні дані:

* кількість відпрацьованих годин за день (ціле, число більше або дорівнює 0 і водночас менше за 24) у год.

Вихідні дані:

* нарахована сума З/П працівника (дійсне, додатне число з точністю 2 знака після коми) у грн.
* вираховані суми податків (дійсне, додатне число з точністю 2 знака після коми) у грн.
* З/П з врахування податків (дійсне, додатне число з точністю 2 знака після коми) у грн.

### **Проектування архітектури програмн**ого **модул**я**:**

Згідно з вимог проектування, реалізуємо функцію, що в якості аргументу приймає масив із 5 чисел, що визначені кількістю відпрацьованих годин за кожний робочий день тижня відповідно. Повертає структуру (оголошення структури визначено в заголовковому файлі) З/П до/після впливу податків, та сума кожного податку.

У випадку, якщо вхідне значення не відповідає вимогам, воно ініціалізується нулем.

Надалі, в циклі з п'ятьма ітераціями розраховується оплата за години праці. Після чого - кожен елемент структури (для цієї задачі) ініціалізується за формулами, що визначають частки податків та отриману З/П.

### **Завдання 9.2**

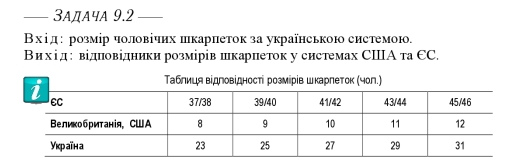


Рисунок 1.2 - Умова задачі 9.2

### **Строга постановка задачі:**

Вхідні дані:

* розмір чоловічих шкарпеток за українською системою (цілочисельне, додатне число з таблиці в умові завдання)

Вихідні дані:

* відповідники розмірів за системою ЄС та США (цілочисельні, додатні числа з таблиці в умові завдання)

### **Проектування архітектури програмного модуля:**

Функція приймає аргумент - розмір чоловічих шкарпеток за укр. системою і повертає структуру відповідників за іншими системами.

У випадку, якщо вхідне значення нерівне одному із вибірки істинних - функція повертає структуру, кожне значення якої визначене як -1. Далі, кожен елемент структури ініціалізується формулою, що залежить від вхідного значення.

### **Завдання 9.3**

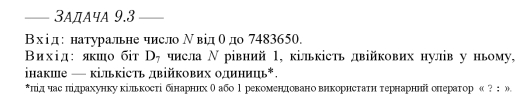


Рисунок 1.3 - Умова задачі 9.3

### **Строга постановка задачі:**

Вхідні та вихідні дані вичерпно описані в умові завдання.

### **Проектування архітектури програмного модуля:**

Реалізована процедура приймає в якості аргументу число та повертає кількість бітів що не рівні 7-му біту цього числа.

Якщо вхідне значення за межами допустимого діапазону, відбувається повернення -1.

Далі, для подальшого опрацювання числа, воно ініціалізується оберненим. В циклі в 31 ітерацію, до змінної, що зберігає кількість бітів, додається результат логічного виразу, що перевіряє відповідний біт та його рівність з 7-м бітом числа. За допомогою оператору множинного вибору, що опрацьовує значення 7-го біта, у відповідності до цього ж значення і обирається виведення кількості необхідних бітів.

### **Завдання 9.4**

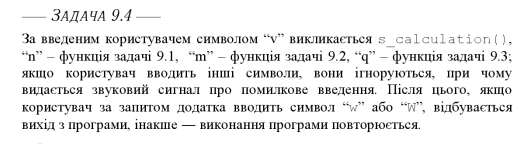


Рисунок 1.4 - Умова задачі 9.3

### **Строга постановка задачі:**

Вхідні дані:

* символьний літерал (із зазначеного переліку), що вводить користувач.

Вихідні дані:

* результат виконання, відповідної до виклику літералом, функції.

### **Проектування архітектури програмного модуля:**

Оголошуемо змінну, що буде генерувати випадкове число та оголошуемо структури, попередньо оформлені в завданнях 9.1 та 9.2. Використовуючи оператор множинного вибору, що обробляє значення, введене з клавіатури, реалізуємо різноманітні шляхи виконання функції. Шляхи виконання визначені в умові, як і літерал, введення якого обумовлює вихід з вічного циклу.

### **Результат запуску застосунку TestDriver**

========================================================================

ПН: 8

ВТ: 8

СР: 6

ЧТ: 1

ПТ: 8

Нарахована сума зарплатні: 1240 грн

Податок на прибуток фізособи (15%): 186 грн

Податок до пенсійного фонду (2%): 24.8 грн

Податок до фонду страхування на випадок безробіття (0.6%): 7.44 грн

Податок на соціально страхування у випадку втрати працезатності (1%): 12.4 грн

Сумма до видачі: 1009.36 грн

Passed

========================================================================

ПН: 0

ВТ: 0

СР: 0

ЧТ: 0

ПТ: 0

Нарахована сума зарплатні: 0 грн

Податок на прибуток фізособи (15%): 0 грн

Податок до пенсійного фонду (2%): 0 грн

Податок до фонду страхування на випадок безробіття (0.6%): 0 грн

Податок на соціально страхування у випадку втрати працезатності (1%): 0 грн

Сумма до видачі: 0 грн

Passed

========================================================================

ПОМИЛКА: від'ємне значення зарплати ! (автоматично визначається як 0)

ПН: 9

ВТ: 0

СР: 9

ЧТ: 9

ПТ: 11

Нарахована сума зарплатні: 2720 грн

Податок на прибуток фізособи (15%): 408 грн

Податок до пенсійного фонду (2%): 54.4 грн

Податок до фонду страхування на випадок безробіття (0.6%): 16.32 грн

Податок на соціально страхування у випадку втрати працезатності (1%): 27.2 грн

Сумма до видачі: 2214.08 грн

Passed

Розмір шкарпеток за українською системою: 25

Розмір шкарпеток за системою США: 10

Розмір шкарпеток за системою ЄС: 39//40

Passed

Розмір шкарпеток за українською системою: 27

Розмір шкарпеток за системою США: 11

Розмір шкарпеток за системою ЄС: 41//42

Passed

Розмір шкарпеток за українською системою: 29

Розмір шкарпеток за системою США: 12

Розмір шкарпеток за системою ЄС: 43//44

Passed

Кількість двійкових 1 = 2

Passed

Кількість двійкових 0 = 9

Passed

Кількість двійкових 1 = 13

Passed

# ВИСНОВКИ

Одним із спостережень виконання лабораторної роботи є факт вирішення задач із попередніх лабораторних, проте з використанням модульної парадигми та більш ширшого спектру взаємодії з середовищем розробки. Керуючись строгою постановкою задачі, попередньо було виявлено та ліквідовано можливі проблеми:

* Слушною думкою було попереднє визначення кодування файлу вихідного коду (UTF-8, адже с цим кодуванням, на відміну від CP-1251 та CP-866 не виникає проблем з відображенням файлу коду на сторінці GitHub);
* Також, передбачено критичне завершення програми (або - виведення результату в консоль) у випадку отримання не коректних вхідних даних. Так в окремих модулях, хибні значення визначаються 0 (завдання 9.1: нулем визначається час роботи), або ж як -1 (в інших завданнях, окрім 9.4), що одразу свідчить про порушення умов алгоритмізації та визначення результату TestCase як Failed.

Виходячи з технічних проблем, підключення стандартних динамічних бібліотек компілятора MinGW через налаштування проекту Code::Blocks було опущено. Необхідні для роботи застосунків бібліотеки переміщені в директорію Software.

Для завдання 9.1 прийняті декілька рішень, щодо оформлення програмного коду:

По-перше - створено окрему структуру, котра зберігає суму З/П на різних етапах та суми всіх податків, залежно від З/П. Тип повернення функції є типом даної структури. Прибігання до цього викликане тим, що при майбутньому тестуванні модуля TestDriver матиме доступ одразу до декількох закритих змінних структури.

По-друге, для випадку, коли в аргументі-масиві присутній елемент з від’ємним значенням, його буде переініціалізовано нулем заради запобігання не коректних результатів виконання програми. В консоль виводиться сповіщення щодо цього випадку.

Аналогічно до задачі 9.1, функція для завдання 9.2 має повертати тип типу структури, котра є завчасно створеною та зберігає в собі всі (дані в умові) відповідники розмірів. Також, якщо отриманий розмір (в якості аргументу) не відноситься до жодного із зазначених до української системи розмірів - функція повертає структуру, кожне значення якої визначене як -1 (що автоматично визначає результат TestCase як Failed). Проектне рішення, щодо оптимізації коду для даної задачі: за таблицею відповідників розмірів була визначена закономірність і з метою запобігання надлишкових рядків коду, змінні міжнародних відповідників були ініцалізовані формулами залежності від українського розміру.

Передбачено отримання хибних вхідних даних і для функції задачі 9.3: повернення в результат -1, що автоматично визначає хибний результат TestCase. Згідно з того, що зчитування бітів за стандартом відбувається з кінця числа - число було попередньо інвертоване. Надалі реалізовано фрагмент коду, що безпосередньо відповідає за побітові маніпуляції:

Всередині циклу, ітерованим 32 рази (дане рішення викликане тим, що число типу int займає саме 32 біти) зберігаємо останнє входження біта 1. Це необхідно для подальшого відсікання “зайвих” нулів з лічильника їх кількості. Після цього, до змінної кількості бітів додається результат логічного порівняння рівності поточного біту з бітом, заданим в умові задачі (інакше кажучи, якщо істина - то до лічильника додається 1, якщо хиба - 0 або нічого). Виведення результату реалізовано за допомогою оператору множинного вибору switch, що перевіряє значення N біту.

Так як для завдання 9.4 попередньо необхідно мати відлагоджений код проекту статичної бібліотеки та згідно з того що кожне вхідне значення відноситься до числового типу даних, в даному завданні на вхід аргументам функцій надсилається випадково згенероване значення. Для зручності використання та для коректної реалізації виходу із програми, виконано наступну маніпуляцію: оператор множинного вибору switch, що приймає символ введений з клавіатури, поміщений у вічний цикл. Для виходу з циклу необхідно ввести символ, що зазначено в умові завдання. Вихід буде виконано не стандартним завершенням циклу, а напряму буде завершено роботу застосунку.

Загалом: лабораторна робота дозволила повноцінно закріпити навички використання статичних бібліотек та врахування всіх аспектів пов’язаних з ними. Паралельно, покращене розуміння маніпулювання бітами певного числа та роботу з функціями що в якості аргументів і/або результату повернення використовують змінні складених типів (структури, масиви, і. т. д.). Врахований факт необхідності кодування файлу, що надає можливість до коректного перегляду його вмісту на сторінці ресурсу GitHub.

**ДОДАТОК А**

(Вихідний код ModulesStriuk)

**… / ModulesStriuk / main.cpp**

#include <iostream>

#include <cmath>

#include "ModulesStriuk.h"

using namespace std;

double s\_calculation(float x, float y, float z)

{

const double PI = 3.14159;

return log(x - y) + sqrt((PI \* pow(x, 2)) / x + (z / (2 \* pow(y, 2))));

}

money salaryAndTaxesOutput(int workingHours[5])

{

cout << "========================================================================" << endl;

/// Перевірка на правильність вказання часу

for (int i = 0; i < 5; i++) {

if (workingHours[i] < 0) {

cout << endl << "ПОМИЛКА: від\'ємне значення зарплати ! (автоматично визначається як 0)" << endl << endl;

workingHours[i] = 0;

}

}

cout << "ПН: " << workingHours[0] << endl

<< "ВТ: " << workingHours[1] << endl

<< "СР: " << workingHours[2] << endl

<< "ЧТ: " << workingHours[3] << endl

<< "ПТ: " << workingHours[4] << endl << endl;

/// Нарахування ЗП згідно з відпрацьованих годин

int fullSalary = 0;

for (int i = 0; i < 5; i++) {

if (workingHours[i] <= 0) { continue; }

else if (workingHours[i] <= 8) { fullSalary += workingHours[i] \* 40; }

else { fullSalary += workingHours[i] \* 40 + 300; }

}

/// Визначення податків та ЗП у кінечному результаті

money m;

m.salary = fullSalary;

m.fizosoba = fullSalary \* 0.15;

m.pensiya = fullSalary \* 0.02;

m.bezrob = fullSalary \* 0.006;

m.pratsa = fullSalary \* 0.01;

m.completeSalary = fullSalary - (fullSalary \* 0.186);

cout << "Нарахована сума зарплатні: "

<< m.salary << " грн" << endl;

cout << "Податок на прибуток фізособи (15%): "

<< m.fizosoba << " грн" << endl

<< "Податок до пенсійного фонду (2%): "

<< m.pensiya << " грн" << endl

<< "Податок до фонду страхування на випадок безробіття (0.6%): "

<< m.bezrob << " грн" << endl

<< "Податок на соціально страхування у випадку втрати працезатності (1%): "

<< m.pratsa << " грн" << endl;

cout << "Сумма до видачі: " << m.completeSalary << " грн" << endl << endl << endl;

return m;

}

socks socksSizeStandart(unsigned char sSize)

{

/// Заверешення функції при недопустимих вхідних даних

if (sSize != 23 && sSize != 25 && sSize != 27 && sSize != 29 && sSize != 31) {

cout << "ПОМИЛКА: Розмір шкарпеток не відповідає стандарту !" << endl;

return {-1, -1, -1, -1};

}

/// Виведення аналогів розміру (для спрощення коду, за знайденною закономірністю, використовуються формули)

socks s;

s.ukrSize = sSize;

s.usSize = (sSize / 2) - 2;

s.esSize1 = sSize + 14;

s.esSize2 = sSize + 15;

cout << "Розмір шкарпеток за українською системою: " << s.ukrSize << endl;

cout << "Розмір шкарпеток за системою США: " << s.usSize << endl;

cout << "Розмір шкарпеток за системою ЄС: " << s.esSize1 << "//" << s.esSize2 << endl;

return s;

}

int numberByteManipulation(int number)

{

/// Завершення функції, при невідповідних до заданого діапазону чисел

if (number < 0 || number > 7483650) {

cout << "ПОМИЛКА: число в недопустимому діапазоні !" << endl;

return -1;

}

int reverseNumber = 0;

int bitCount = 0;

/// Інвертування числа (біти, за замовчуванням, рахуються з кінця числа)

for (; number; number /= 2) {

reverseNumber = reverseNumber \* 2 + number % 2;

}

number = reverseNumber;

/// Додавання до кількості бітів результату логічного виразу (біт не має бути рівним сьомому)

int saveIndex = 0;

for (int i = 0; i <= 31; i++) {

if (number & (1 << i)) { saveIndex = i + 1; }

bitCount += ((number & (1 << i)) != ((number >> 7) & 1));

}

if ((number >> 7) & 1) { bitCount = bitCount - (31 - saveIndex); }

switch ((number >> 7) & 1) {

case 0: cout << "Кількість двійкових 1 = " << bitCount << endl; break;

case 1: cout << "Кількість двійкових 0 = " << bitCount << endl; break;

}

return bitCount;

}

**… / ModulesStriuk / ModulesStriuk.h**

#ifndef MODULESSTRIUK\_H\_INCLUDED

#define MODULESSTRIUK\_H\_INCLUDED

struct money {

float salary;

float fizosoba;

float pensiya;

float bezrob;

float pratsa;

float completeSalary;

};

struct socks {

int ukrSize;

float usSize;

int esSize1;

int esSize2;

};

double s\_calculation(float x, float y, float z);

money salaryAndTaxesOutput(int workingHours[5]);

socks socksSizeStandart(unsigned char ukrSoksSize);

int numberByteManipulation(int number);

#endif // MODULESSTRIUK\_H\_INCLUDED

**ДОДАТОК Б**

(Вихідний код TestDriver)

**… / TestDriver / main.cpp**

#include <iostream>

#include "ModulesStriuk.h"

using namespace std;

int main()

{

system("chcp 65001 & cls");

money m;

int workingHours\_1[5] {8, 8, 6, 1, 8};

int workingHours\_2[5] {0, 0, 0, 0, 0};

int workingHours\_3[5] {9, -9, 9, 9, 11};

m = salaryAndTaxesOutput(workingHours\_1);

if (m.salary\*100 == 124000 && m.fizosoba\*100 == 18600 &&

m.pensiya\*100 == 2480 && m.bezrob\*100 == 744 &&

m.pratsa\*100 == 1240 && m.completeSalary\*100 == 100936) {

cout << "Passed" << endl;

}

else { cout << "Failed" << endl;}

m = salaryAndTaxesOutput(workingHours\_2);

if (m.salary\*100 == 0 && m.fizosoba\*100 == 0 &&

m.pensiya\*100 == 0 && m.bezrob\*100 == 0 &&

m.pratsa\*100 == 0 && m.completeSalary\*100 == 0) {

cout << "Passed" << endl;

}

else { cout << "Failed" << endl;}

m = salaryAndTaxesOutput(workingHours\_3);

if (m.salary\*100 == 272000 && m.fizosoba\*100 == 40800 &&

m.pensiya\*100 == 5440 && m.bezrob\*100 == 1632 &&

m.pratsa\*100 == 2720 && m.completeSalary\*100 == 221408) {

cout << "Passed" << endl;

}

else { cout << "Failed" << endl;}

int socksSize[3] = {25, 27, 29};

cout << endl << endl << endl << endl;

for (int i = 0; i < 3; i ++) {

socks s = socksSizeStandart(socksSize[i]);

if (s.ukrSize == socksSize[i] && s.usSize == (socksSize[i] / 2) - 2 &&

s.esSize1 == socksSize[i] + 14 && s.esSize2 == socksSize[i] + 15) {

cout << "Passed" << endl;

}

else { cout << "Failed" << endl; }

cout << endl;

}

int numberEx[3] = {12, 678, 3938683};

int bitEx[3] = {2, 9, 13};

for (int i = 0; i < 3; i++) {

if (numberByteManipulation(numberEx[i]) == bitEx[i]) {

cout << "Passed" << endl;

}

else { cout << "Failed" << endl; }

}

getchar();

return 0;

}

**ДОДАТОК В**

(Вихідний код Striuk\_task)

**… / Striuk\_task / main.cpp**

#include <iostream>

#include <conio.h>

#include <random>

#include <time.h>

#include "ModulesStriuk.h"

using namespace std;

int main()

{

system("chcp 65001 & cls");

srand(time(0));

money m;

socks s;

int bitNumber;

while(true) {

cout << "Введіть символ (v/n/m/q/w/W): ";

int salaryArray[5] = {rand()%24, rand()%24, rand()%24, rand()%24, rand()%24};

int randNum;

switch (\_getch()) {

case 'v': cout << endl << endl << "s\_calculation: " << s\_calculation(rand(), rand(), rand()) << endl; break;

case 'n': cout << endl << "salaryAndTaxesOutput: " << endl;

m = salaryAndTaxesOutput(salaryArray);

break;

case 'm': randNum = rand()%35;

cout << endl << endl << "socksSizeStandart: " << randNum << endl;

s = socksSizeStandart(randNum);

break;

case 'q': randNum = rand();

cout << endl << endl << "numberByteManipulation: " << randNum << endl;

bitNumber = numberByteManipulation(randNum);

break;

case 'w': return 0; break;

case 'W': return 0; break;

default: cout << "\a" << endl; continue;

}

}

}

**ДОДАТОК Г**

(TestSuite до ModulesStriuk)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_MODULE |
| Date: 03/18/2022 | Назва проекта/ПЗ /  **Name of project** | TestDriver.ехе |
|  | Рівень тестування /  **Level of testing** | Модульний |
|  | Автор тест-сьюта /  **Test Suite Author** | Владислав Стрюк |
|  | Виконавець /  **Implementer** | Владислав Стрюк |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Actions** | **Expected Result** | **Test Result** |
| TS1\_01 | workingHours\_1 = {8, 8, 6, 1, 8} | salary=1240,00  fizosoba=186,00  pensiya=24,80  bezrob=7,44  pratsa=12,40  completeSalary=1009,36 | Passed |
| TS1\_02 | workingHours\_2 = {0, 0, 0, 0, 0} | salary=0,00  fizosoba=0,00  pensiya=0,00  bezrob=0,00  pratsa=0,00  completeSalary=0,00 | Passed |
| TS1\_03 | workingHours\_3 = {9, -9, 9, 9, 11} | salary=2720,00  fizosoba=408,00  pensiya=54,40  bezrob=16,32  pratsa=27,20  completeSalary=2214,08 | Passed |
| TS2\_01 | socksSize = 25 | usSize = 10  esSize1 = 39  esSize2 = 40 | Passed |
| TS2\_02 | socksSize = 27 | usSize = 11  esSize1 = 41  esSize2 = 42 | Passed |
| TS2\_03 | socksSize = 29 | usSize = 12  esSize1 = 43  esSize2 = 44 | Passed |
| TS3\_01 | numberEx= 12 | bitEx = 2 | Passed |
| TS3\_02 | numberEx= 678 | bitEx = 9 | Passed |
| TS3\_03 | numberEx= 3938683 | bitEx = 13 | Passed |

**ДОДАТОК Д**

(TestSuite до Striuk\_task)

|  |  |  |
| --- | --- | --- |
| Artifact: Test Suite | Назва тестового набору /  **Test Suite Description** | TS\_9\_4 |
| Date: 03/18/2022 | Назва проекта/ПЗ /  **Name of project** | Striuk-task.ехе |
|  | Рівень тестування /  **Level of testing** | Системний |
|  | Автор тест-сьюта /  **Test Suite Author** | Владислав Стрюк |
|  | Виконавець /  **Implementer** | Владислав Стрюк |

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест кейса /  **Test Case ID** | Дії (кроки)  /  **Action (Test Steps)** | Очікуваний результат  /  **Expected Result** | Результат тестування  /  **Test Result** |
| TS\_01 | 1. Відкрити застосунок | Введіть символ (v/n/m/q/w/W): | Passed |
| TS\_02 | 1. Ввести ‘o’  2. Натиснути Enter | Введіть символ (v/n/m/q/w/W):  Введіть символ (v/n/m/q/w/W):  Введіть символ (v/n/m/q/w/W): | Passed |
| TS\_03 | 1. Ввести ‘v’ | s\_calculation: <результат> | Passed |
| TS\_04 | 1. Ввести ‘n’ | salaryAndTaxesOutput:  ПН: 13  ВТ: 9  СР: 18  ЧТ: 2  ПТ: 10  Нарахована сума зарплатні: 3280 грн  Податок на прибуток фізособи (15%): 492 грн  Податок до пенсійного фонду (2%): 65.6 грн  Податок до фонду страхування на випадок безробіття (0.6%): 19.68 грн  Податок на соціально страхування у випадку втрати працезатності (1%): 32.8 грн  Сумма до видачі: 2669.92 грн | Passed |
| TS\_05 | 1. Ввести ‘m’ | socksSizeStandart: 23  Розмір шкарпеток за українською системою: 23  Розмір шкарпеток за системою США: 8  Розмір шкарпеток за системою ЄС: 37//38 | Passed |
| TS\_06 | 1. Ввести ‘m’ | socksSizeStandart: 7  ПОМИЛКА: Розмір шкарпеток не відповідає стандарту ! | Passed |
| TS\_07 | 1. Ввести ‘q’ | numberByteManipulation: 7707  Кількість двійкових 1 = 8 | Passed |
| TS\_08 | 1. Ввести ‘w’ | Завершення роботи застосунку | Passed |