**ЛАБОРАТОРНА РОБОТА № 7-8**

**Програмування розгалужених алгоритмів. Оператори циклу for і while**

***Мета:*** вивчити особливості використання операторів циклу.

**Хід роботи:**

**Завдання 1.** Написати програму:

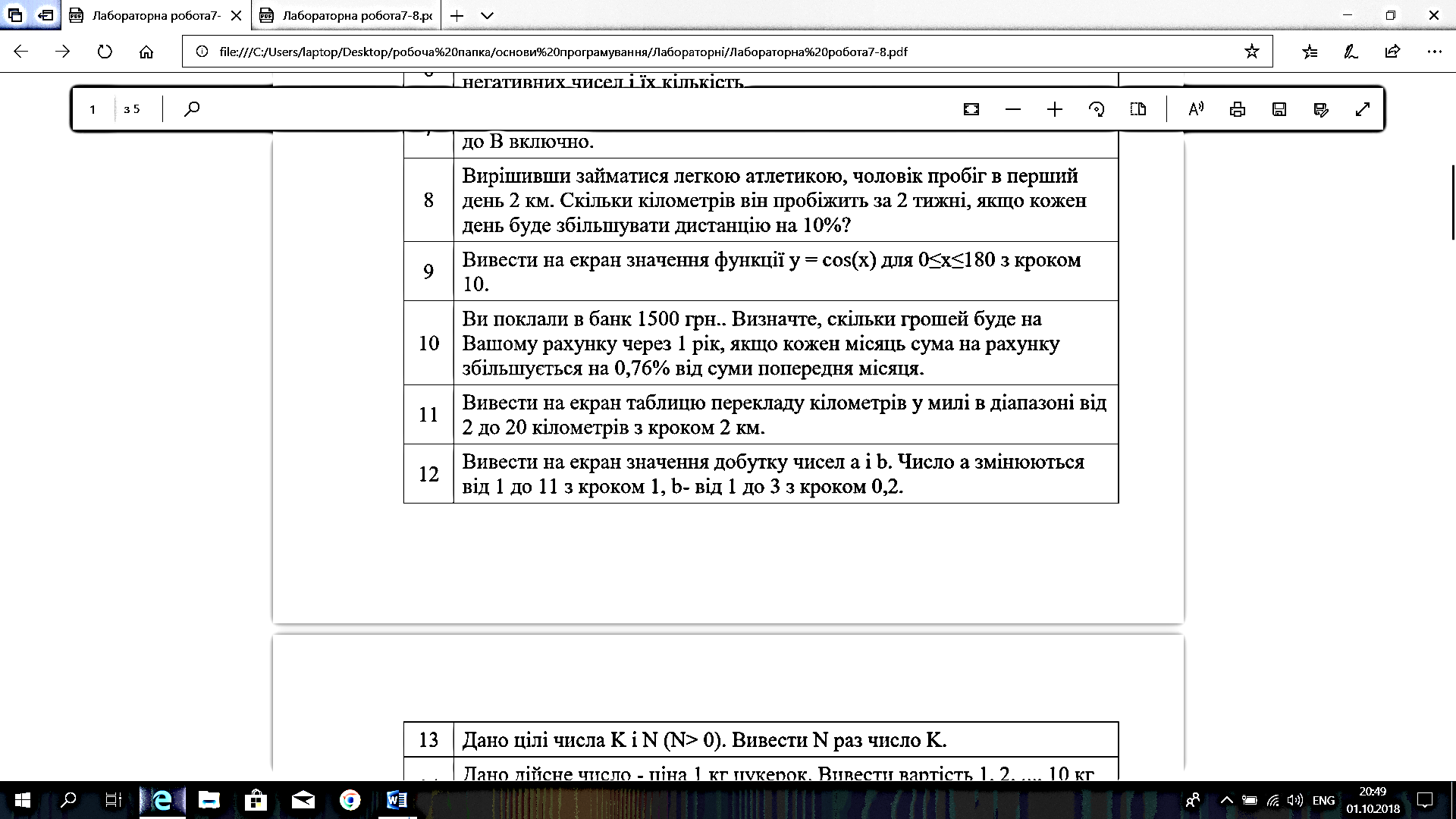


Рис. 1. Умова до завдання № 1

Лістинг програми:

#include "pch.h"

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float i, n, sum = 1500;

printf("Введите кол-во повторений цикла:");

scanf\_s("%f", &n);

for (i = 0; i < n; i++)

sum += 0.76\*sum / 100;

printf("Сумма = %f", sum);

return 0;

}

Результат виконання програми:

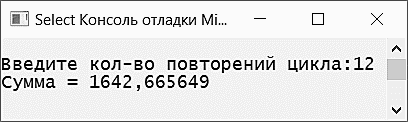


Рис. 2. Результат програми до завдання № 1

Лістинг програми:

#include "pch.h"

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float i = 0, n, sum = 1500;

printf("\n\nВведите кол-во повторений цикла:");

scanf\_s("%f", &n);

while (i < n) {

sum += 0.76\*sum / 100; i++; }

printf("Сумма = %f\n\n", sum);

return 0;

}

Результат виконання програми:

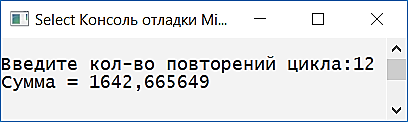


Рис. 3. Результат програми до завдання № 1

**Завдання 2:** Дано натуральне число n . Підрахувати:

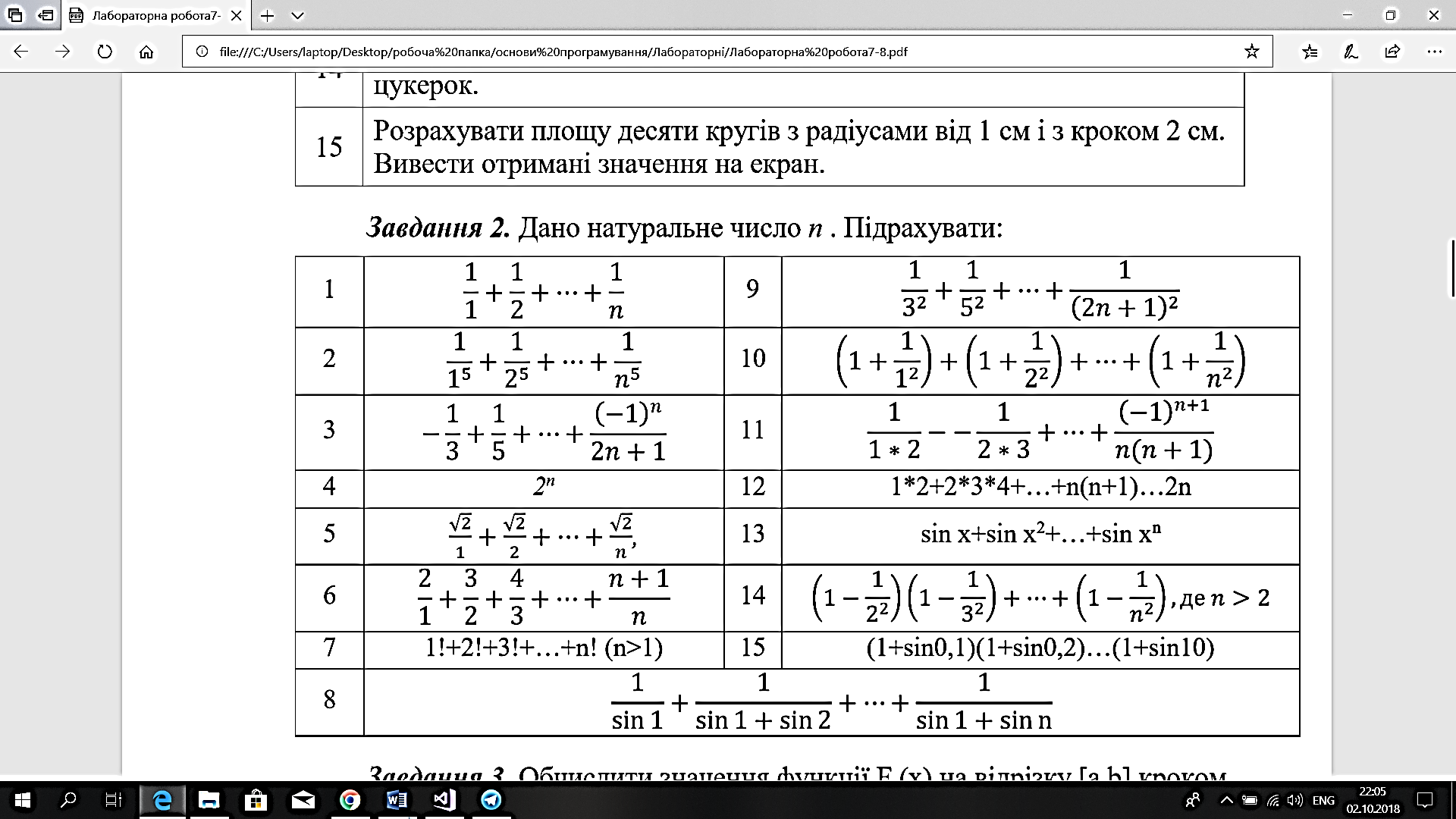


Рис. 4. Умова до завдання № 2

Лістинг програми:

#include "pch.h"

#include <iostream>

#include <locale.h>

#include <math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float n, i, sum = 0;

printf("\n\nВведите кол-во повторений цикла:");

scanf\_s("%f", &n);

for (i = 1; i <= n; i++)

sum = sum + (1 + 1 / pow(i, 2));

printf("Сумма = %2f\n\n", sum);

return 0;}

Результат виконання програми:

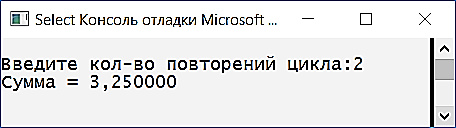


Рис. 5. Результат програми до завдання № 2

Лістинг програми:

include "pch.h"

#include <iostream>

#include <locale.h>

#include <math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float i = 1, n, sum = 0;

printf("\n\nВведите кол-во повторений цикла:");

scanf\_s("%f", &n);

while (i <= n)

{

sum = sum + (1 + 1 / pow(i, 2)); i++;

}

printf("Сумма = %f", sum);

return 0;

}

Результат виконання програми:

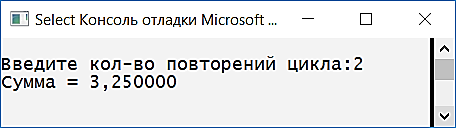


Рис. 6. Результат програми до завдання № 2

**Завдання 3.** Обчислити значення функції F (х) на відрізку [а,b] кроком h.

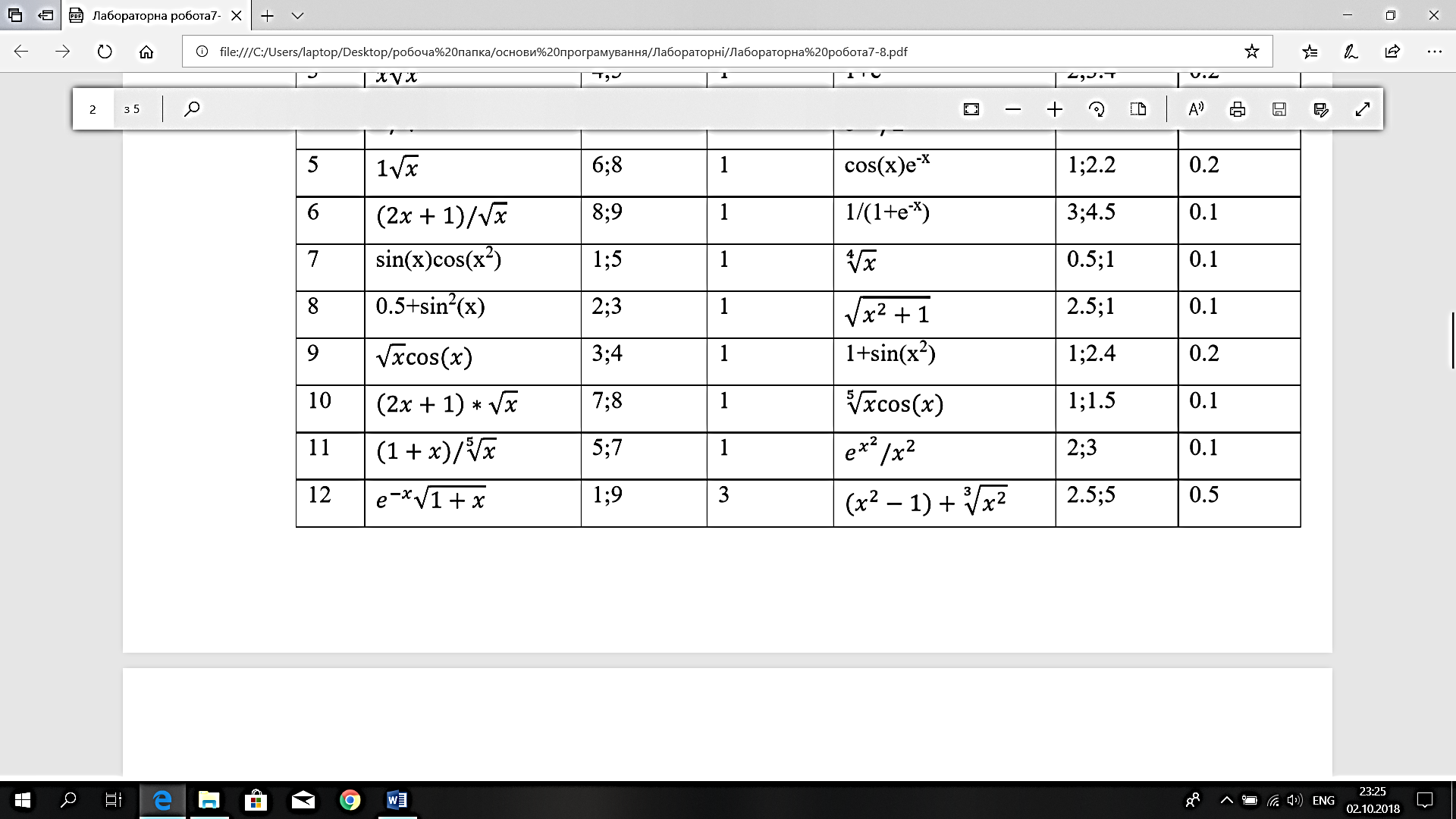
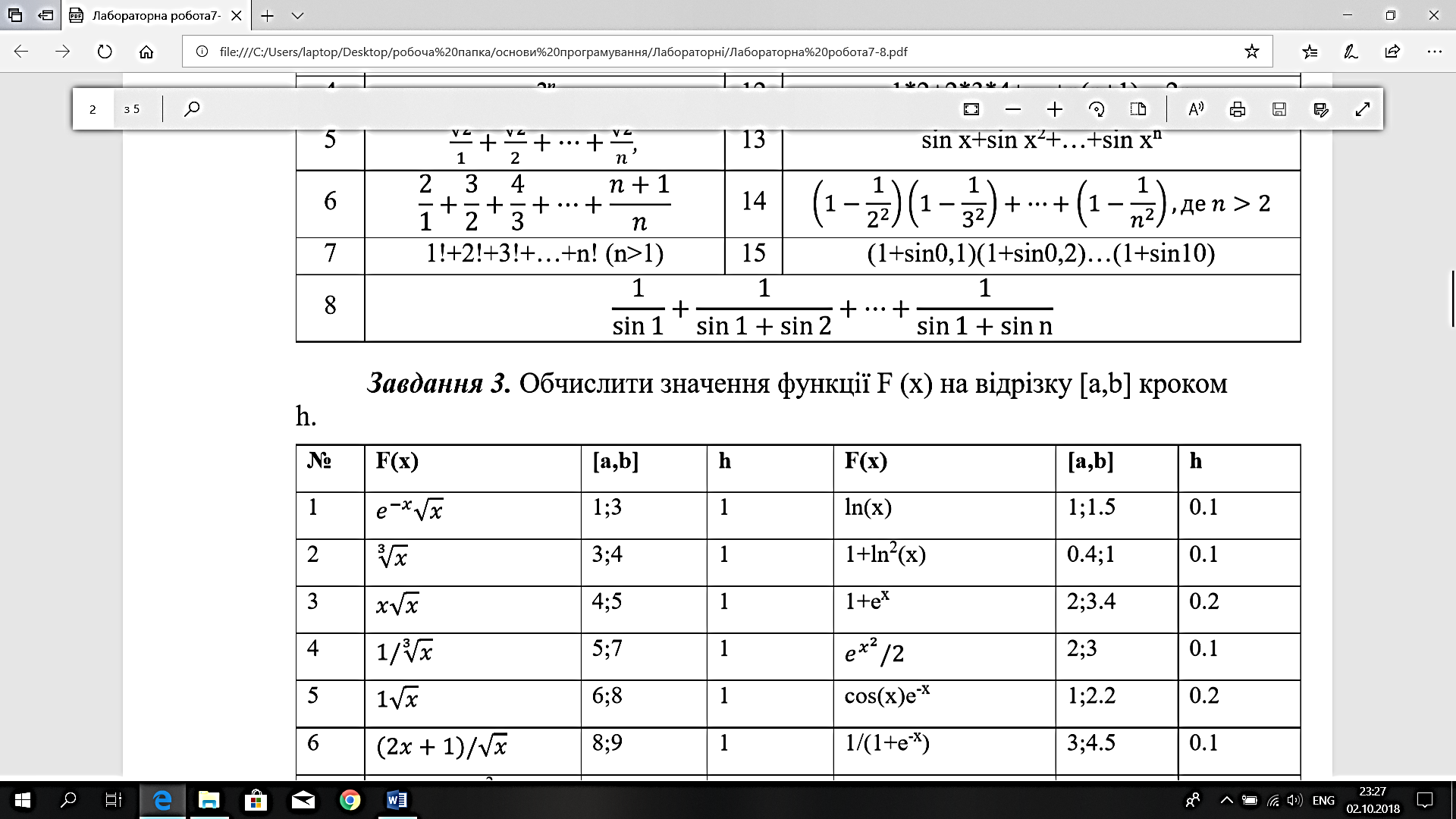


Рис. 7. Умова до завдання № 3

**1.** Лістинг программи:  
#include "pch.h"

#include <iostream>

#include <locale.h>

#include<math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float a = 7, b = 8, h = 1, i, result;

for (i = 7; a <= i && i <= b; i+=h) {

result = (2 \* i + 1)\*sqrt(i);

printf("Значение функции = %f\n\n", result);

}

}

Результат виконання програми:

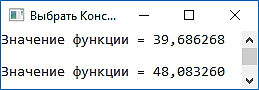


Рис. 8. Результат програми до завдання № 3 (1)

Лістинг программи:  
#include "pch.h"

#include <iostream>

#include <locale.h>

#include<math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float a = 7, b = 8, h = 1, i=7, result;

while (a <= i && i <= b) {

result = (2 \* i + 1)\*sqrt(i); i += h;

printf("Значение функции = %f\n\n", result);

}

}

Результат виконання програми:

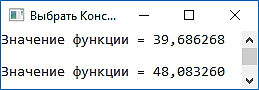


Рис. 9. Результат програми до завдання № 3 (1)

**2.** Лістинг программи:  
#include "pch.h"

#include <iostream>

#include <locale.h>

#include<math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float i, a = 1, b = 1.5, h = 0.1, result = 0, x = 0;

for (i = 1; a <= i && i <= b; i += h) {

x = i \* cos(i);

result = pow(x, 0.2);

printf("Значение функции = %f\n\n", result);

}

}

Результат виконання програми:

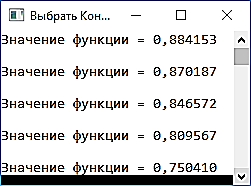


Рис. 10. Результат програми до завдання № 3 (2)

Лістинг программи:  
#include "pch.h"

#include <iostream>

#include <locale.h>

#include<math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

float i = 1, a = 1, b = 1.5, h = 0.1, result = 0, x = 0;

while (a <= i && i <= b) {

x = i \* cos(i); i += h;

result = pow(x, 0.2);

printf("Значение функции = %f\n\n", result);

}

}

Результат виконання програми:

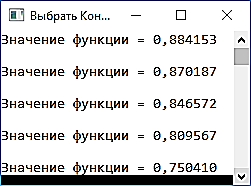


Рис. 11. Результат програми до завдання № 3 (2)

**Завдання на самостійну роботу**

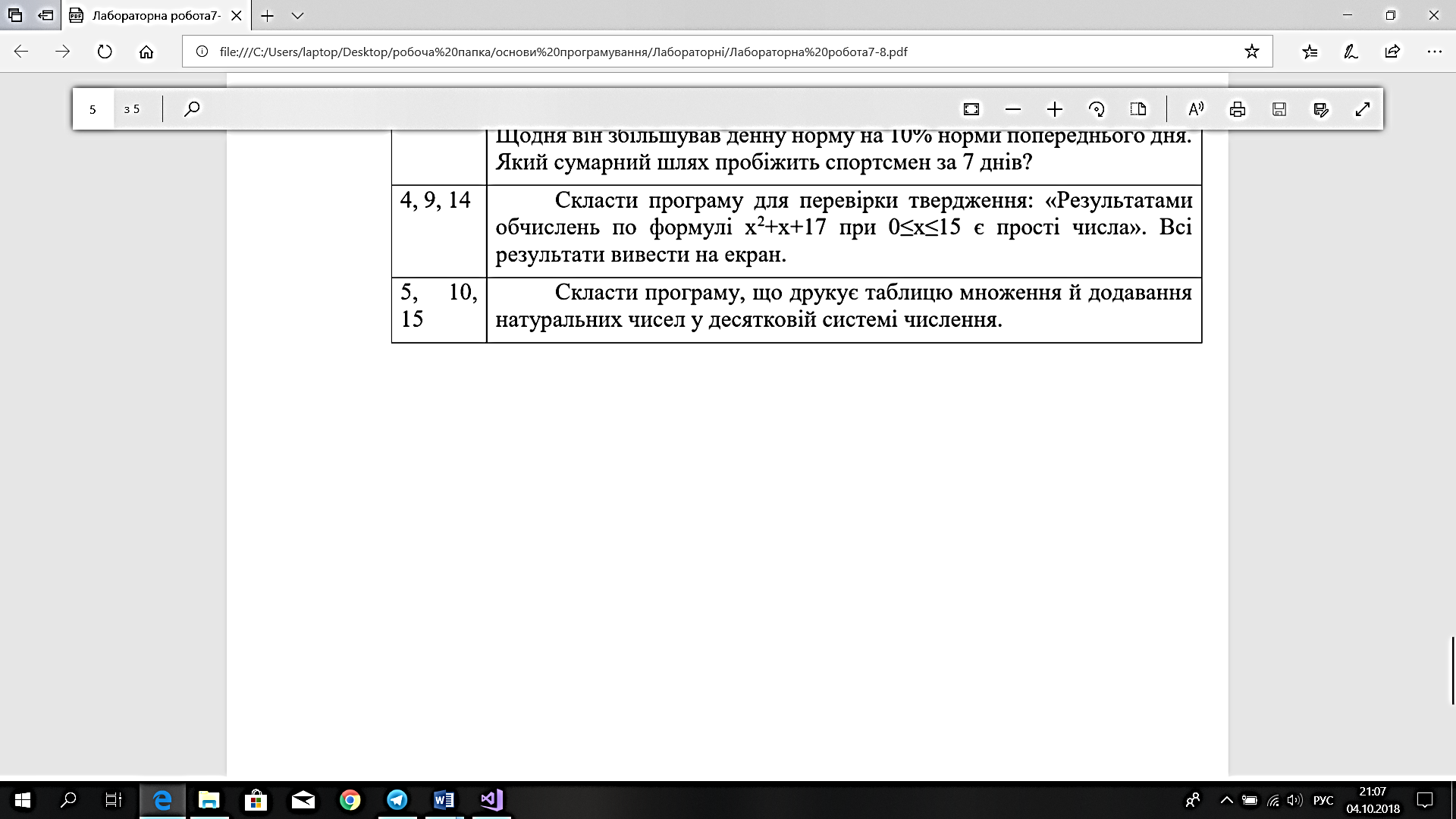


Рис. 12. Умова до завдання № 1

Лістинг програми:

#include "pch.h"

#include <iostream>

#include <locale.h>

#include<math.h>

int main()

{

setlocale(LC\_ALL, "Rus");

int i, j, n = 10, type;

printf("Выберите таблицу: 1 - умножение. 2 - складывание\n");

scanf\_s("%d", &type);

if (type == 1) {

for (i = 1; i <= n; i++) {

for (j = 1; j <= n; j++) {

printf("%d\*%d=%d ", i, j, i\*j); }

printf("\n");

}

}

if (type == 2) {

for (i = 1; i <= n; i++) {

for (j = 1; j <= n; j++) {

printf("%d+%d=%d ", i, j, i + j); }

printf("\n");

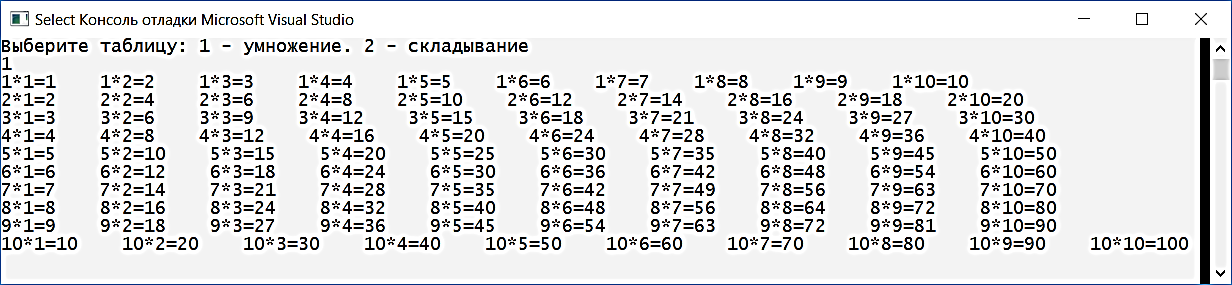
}

}

return 0;

}

Результат виконання програми:



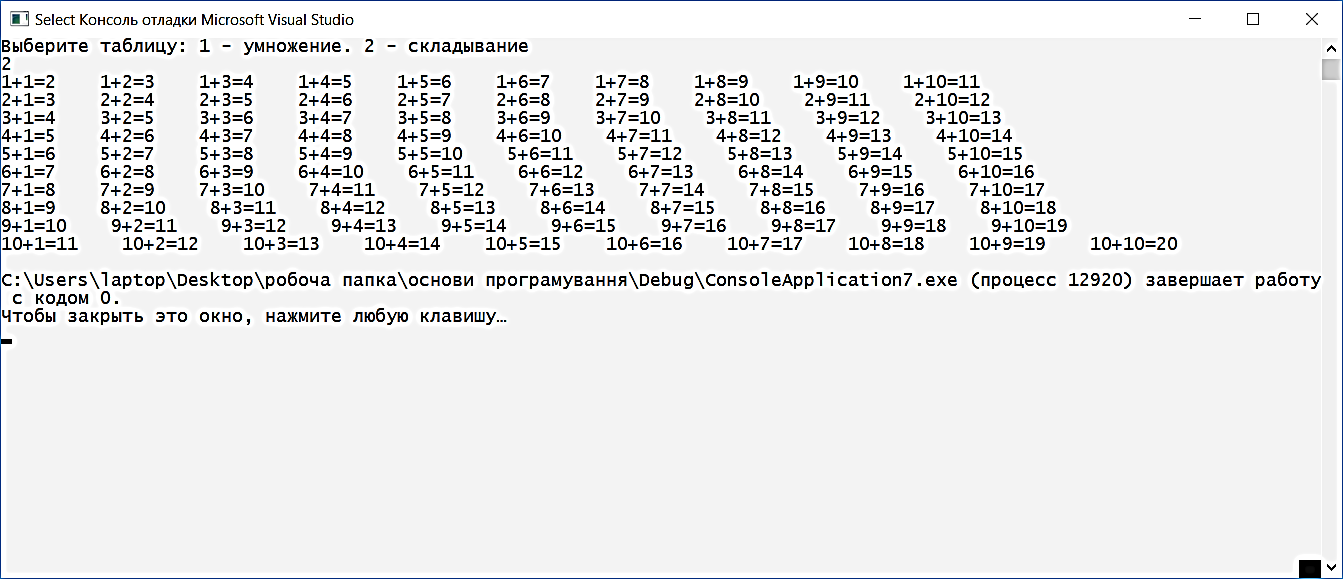


Рис. 13. Результат програми до завдання на самостійну роботу.

**Контрольні питання:**

1. Для яких цілей використовуються цикли в програмуванні?

Для організації багаторазового виконання набору інструкцій (команд) або багатократного виконування послідовністі команд, організованих будь-яким чином (наприклад, із допомогоюумовного переходу).

1. Перерахуйте оператори циклів в мові Сі.

Види операторів: цикл for; цикл while з передумовою; цикл do…while з постумовою.

1. Що таке цикл з передумовою?

Цикл з передумовою — цикл, що виконується доки істинна деяка умова, вказана перед його початком. Ця умова перевіряється до початку виконання тіла циклу, тому тіло може бути не виконане жодного разу (якщо умова з початку хибна).

1. Опишіть синтаксис оператору for.

У круглих дужках міститься три вирази. Перше з них служить для ініціалізації лічильника. Вона здійснюється тільки один раз - коли цикл for починає виконуватися. Друге вираження необхідно для перевірки умови, яка здійснюється перед кожним можливим виконанням тіла циклу. Коли вираз стає помилковим, цикл завершується. Третє вираз обчислюється в кінці кожного виконання тіла циклу, відбувається прирощення числа на крок. Цикл for виконується до тих пір, поки значення вираз рівне true. Як тільки значення вираз стане false, виконання циклу припиняється і виконується оператор, що слідує за циклом for.

1. Опишіть синтаксис оператору while.

Виконується оператор до тих пір, поки значення виразу в дужках правдиве. Перевірка значення виразу відбувається перед кожним виконанням оператора. Коли значення виразу помилково, цикл while закінчується. Якщо вираз брехливий з самого початку, оператор не виконується жодного разу.

Послідовність операторів виконується до тих пір, поки умовний вираз повертає значення true. Як тільки вираз стає рівним false, виконання циклу while припиняється і управління передається наступному за циклом while оператору.

1. У чому полягають відмінності між операторами for і while.

Оператор, який знаходиться у циклі for містить три вирази. Перше з них служить для ініціалізації лічильника. Вона здійснюється тільки один раз - коли цикл for починає виконуватися. Друге вираження необхідно для перевірки умови, яка здійснюється перед кожним можливим виконанням тіла циклу. Коли вираз стає помилковим, цикл завершується.

Оператор, який знаходиться у циклі while виконується до тих пір, поки значення виразу в дужках правдиве. Перевірка значення виразу відбувається перед кожним виконанням оператора. Коли значення виразу помилково, цикл while закінчується. Якщо вираз брехливий з самого початку, оператор не виконується жодного разу.

1. Які цикли називаються вкладеними?

У деяких випадках важливо повторити підзадачу кілька разів усередині більш загальної задачі. Один зі способів написання такої програми - включити цикл у набір інструкцій, що повторюються всередині іншого циклу. Така структура, що складається з циклу в циклі, називається вкладеними циклами.

1. Для чого служать оператори переривання break та continue? Наведіть приклад.

Оператор break робить примусовий вихід з циклу, навіть коли умова циклу – істинно. Оператор break зручно використовувати, коли необхідно вийти з циклу при особливих обставинах. Оператор continue потрібен якщо необхідно пропустити якийсь блок коду, але при цьому не припиняти роботу циклу.

1. Як можна організувати нескінченні цикли? Наведіть декілька варіантів і поясніть їх.

Нескінчений цикл – це цикл, який ніколи не закінчується.

При програмуванні циклічних процесів, програміст помилково може написати код циклу, який ніколи не закінчується.

Крім того, інколи потрібно, щоб цикли містили код спеціального завершення з допомогою інструкції break.

1. Як можна вийти з нескінченних циклів?

Існує три оператори переривання циклу: break, return і continue. Перші два здатні повністю перервати роботу циклу, а continue перериває лише роботу поточної ітерації.

1. Що відбувається при запуску нескінченного циклу?

Такий цикл виконується без кінця.

1. Чи може оператор циклу не мати тіла? Чому?

Оператор циклу не може не мати тіла, тому що для організації циклічних процесів призначені оператори повтору. Вони вказують на те, що певну групу операторів необхідно виконати декілька разів. Ця група операторів називається тілом циклу.

1. Які оператори використовують для організації розгалуження у програмах?

Для кодування розгалуження, коли вибір напрямку подальших дій визначається у залежності від виконання або невиконання певної умови використовують оператори if (якщо) і else (інакше)

***Висновок***: в ході виконання лабораторної роботи було вивчено особливості використання оператора вибору switch.