

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут» ім. Ігоря Сікорського

# Лабораторна робота №1

***з дисципліни «Паралельне програмування»***

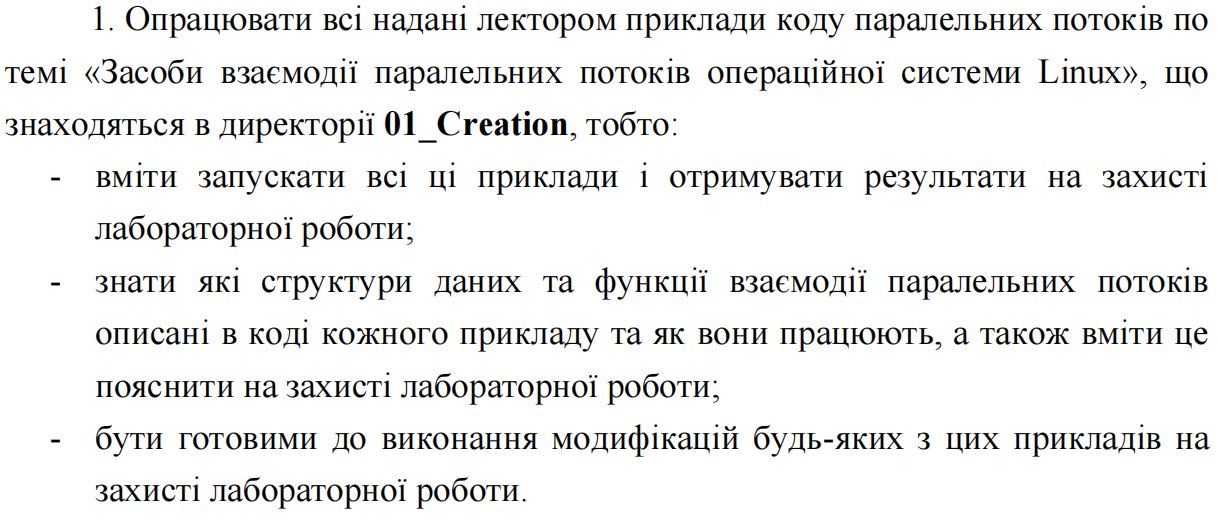
**“*СТВОРЕННЯ ПАРАЛЕЛЬНИХ ПОТОКІВ ОПЕРАЦІЙНОЇ СИСТЕМИ LINUX ТА НАЙПРОСТІША СИНХРОНІЗАЦІЯ*”**

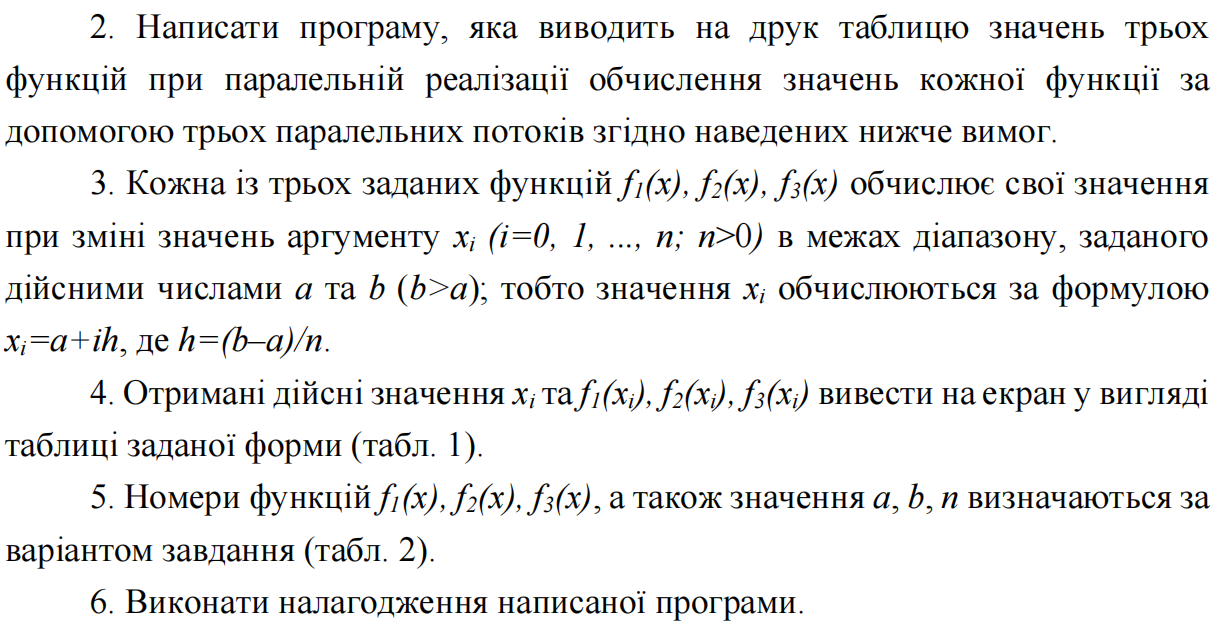
Виконала студентка групи: КВ-31

Кузьменко Олександра

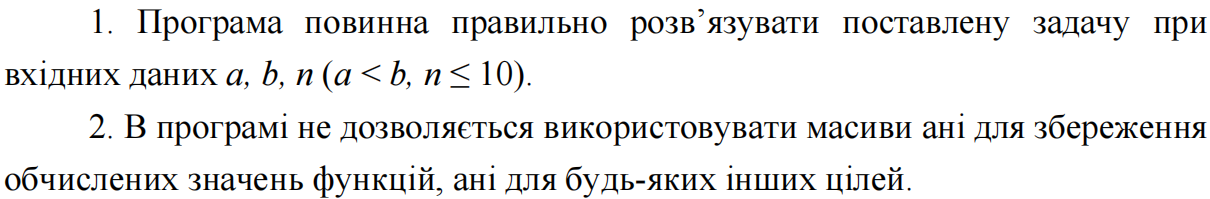
**Київ 2025**

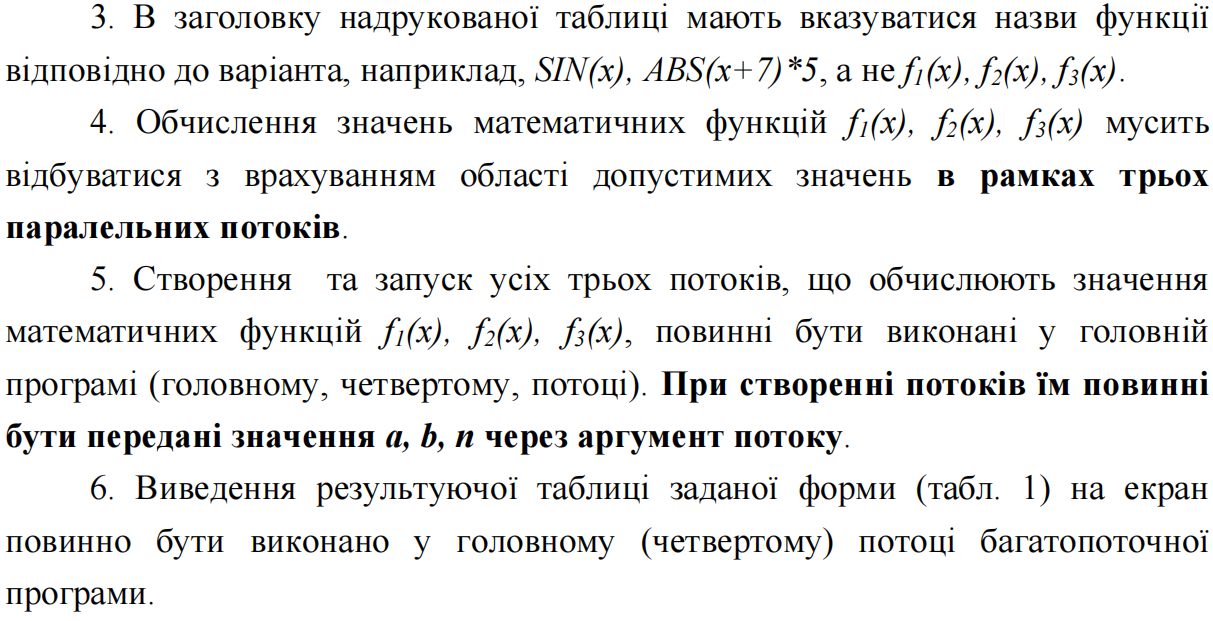
**Постановка задачі**

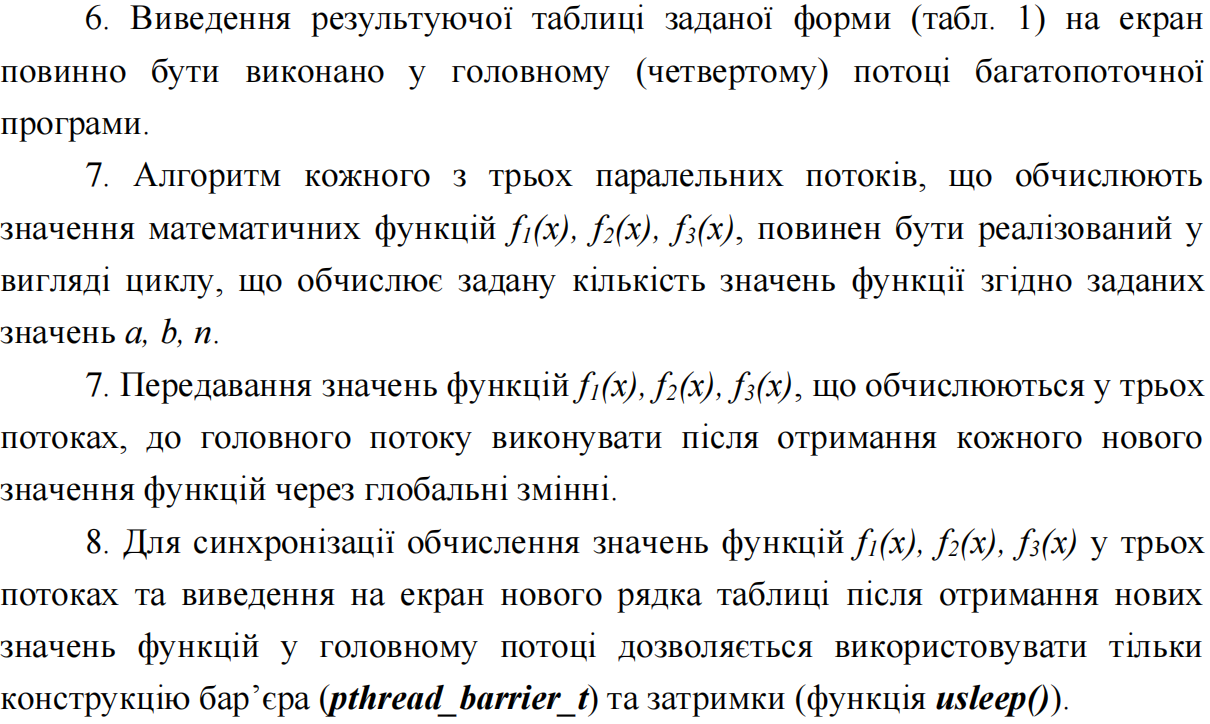




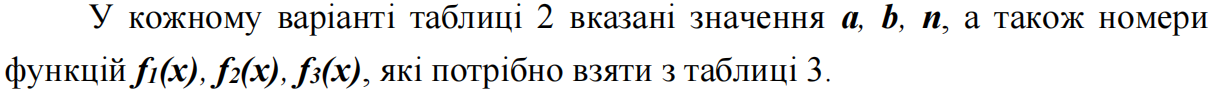
**Вимоги до програми**



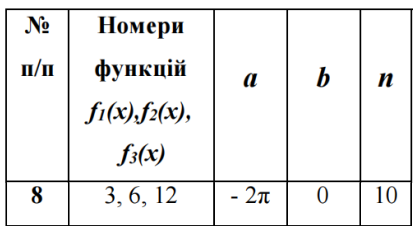
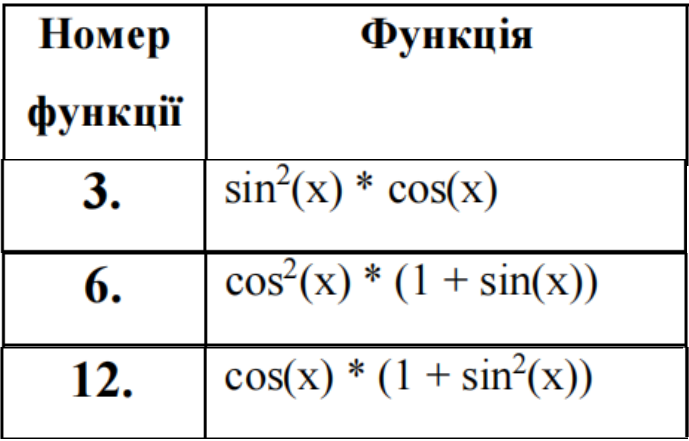




**Завдання за варіантом 8**



Таблиця 2 Таблиця 3



**Код програми**

#include <pthread.h>

#include <stdio.h>

#include <math.h>

float f1r;

float f2r;

float f3r;

pthread\_barrier\_t br1;

struct values

{

float a;

float b;

short n;

float h;

};

void\* fn1(void\* arg)

{

float x;

struct values thread\_arg = \*((struct values\*)arg);

for (short i = 0; i <= thread\_arg.n; i++)

{

x = thread\_arg.a + thread\_arg.h \* i;

f1r = sin(x) \* sin(x) \* cos(x);

pthread\_barrier\_wait(&br1);

pthread\_barrier\_wait(&br1);

}

return NULL;

}

void\* fn2(void\* arg)

{

float x;

struct values thread\_arg = \*((struct values\*)arg);

for (short i = 0; i <= thread\_arg.n; i++)

{

x = thread\_arg.a + thread\_arg.h \* i;

f2r = cos(x) \* cos(x) \* (1 + sin(x));

pthread\_barrier\_wait(&br1);

pthread\_barrier\_wait(&br1);

}

return NULL;

}

void\* fn3(void\* arg)

{

float x;

struct values thread\_arg = \*((struct values\*)arg);

for (short i = 0; i <= thread\_arg.n; i++)

{

x = thread\_arg.a + thread\_arg.h \* i;

f3r = cos(x) \* (1 + sin(x) \* sin(x));

pthread\_barrier\_wait(&br1);

pthread\_barrier\_wait(&br1);

}

return NULL;

}

int main()

{

float a = -2 \* M\_PI;

float b = 0;

short n = 10;

float h = (2 \* M\_PI) / 10;

pthread\_t f1, f2, f3;

struct values data;

data.a = a;

data.b = b;

data.n = n;

data.h = h;

printf("\tx\t| sin^2+cos | cos^2\*(1+sin) | cos\*(1+sin^2) |\n");

printf("----------------|-------------|---------------|---------------|\n");

pthread\_barrier\_init(&br1, NULL, 4);

pthread\_create(&f1, NULL, fn1, &data);

pthread\_create(&f2, NULL, fn2, &data);

pthread\_create(&f3, NULL, fn3, &data);

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

{

float x = a + i \* h;

printf(" %8.5f\t|", x);

pthread\_barrier\_wait(&br1);

printf(" %8.5f |", f1r);

printf(" %8.5f |", f2r);

printf(" %8.5f |\n", f3r);

pthread\_barrier\_wait(&br1);

}

printf("----------------|-------------|---------------|---------------|\n");

pthread\_join(f1, NULL);

pthread\_join(f2, NULL);

pthread\_join(f3, NULL);

pthread\_barrier\_destroy(&br1);

return 0;

}

**Mikefile**

.PHONY: greet build rebuild run clean

greet:

@echo " build : fast rebuild / build"

@echo " rebuild : full rebuild"

@echo " run : run after fast rebuild / build"

@echo " clean : perform full clean"

build: main

rebuild: clean main

run: main

./main

clean:

rm -rvf \*.o main

main: main.c

gcc -o main main.c -pthread -lm

**Таблиця з функціями**

