

Міністерство освіти і науки України  
НТУУ «КПІ ім. Ігоря Сікорського»  
Навчально-науковий інститут атомної та теплової енергетики  
Кафедра цифрових технологій в енергетиці

## Лабораторна робота №3

з дисципліни

«Моделювання систем в енергетиці»

Тема «Розробка моделі розрахунку енергетичних потреб на  
опалення/охолодження будівель»

Варіант №18

Студента 4-го курсу НН ІАТЕ гр. ТР-12

Ковальова Олександра

Перевірила: ст. вик., Висоцька Олена Іванівна

**Завдання:** Створити комп'ютерну програму розрахунку енергетичних потреб житлових приміщень на опалення/охолодження з використання моделі, яка базується на ДСТУ 9190:2022 «Енергетична ефективність будівель»

**Вхідні дані:**

Загалом, додаток може використовуватись в двох сценаріях: базовий та з впровадженням енергозберігаючих заходів. Всі значення можна знайти в описі коду json файлу, який подається на вхід, але також вони описані в цьому розділі:

1. Площа помешкання – 56 метрів квадратних.
2. Стіни:
  - a. Стіна 1 (зовнішня):
    - i. Ширина – 5 метрів;
    - ii. Висота – 2.7 метрів;
    - iii. Шар 1: Цегла керамічна (товщина 0.25 м.);
    - iv. Шар 2: Газо- та пінозолобетон (товщина 0.25 м.);
    - v. Шар 3: Штукатурка вапняна темно-сіра (товщина 0.02 м.);
    - vi. Коефіцієнти теплообміну внутрішньої і зовнішньої поверхонь огорожувальної конструкції – відповідно 8.7 та 23.0 відповідно (згідно таблиці 2 лабораторної роботи);
    - vii. Коефіцієнт поглинання сонячної радіації,  $\alpha_{S,c}$  – 0.7 (Згідно таблиці 4 лабораторної роботи);
    - viii. Коефіцієнт теплового випромінювання зовнішньою поверхнею,  $\varepsilon$  – 0.93 (Згідно таблиці 4 лабораторної роботи).
  - b. Стіна 2 (зовнішня):
    - i. Ширина – 4.0 метрів;
    - ii. Висота – 2.7 метрів;
    - iii. Останні показники такі ж, як і у стіни 1.
  - c. Стіна 3 (зовнішня):
    - i. Ширина – 4.0 метрів;
    - ii. Висота – 2.7 метрів;
    - iii. Останні показники такі ж, як і у стіни 1.
  - d. Стіни 4-5-6 (внутрішні):
    - i. Ширина – 4.0 метрів;
    - ii. Висота – 2.7 метрів;
    - iii. Єдиний шар: Газо- та пінозолобетон (товщина 0.25 м.);
    - iv. Коефіцієнт поглинання сонячної радіації,  $\alpha_{S,c}$  – 0 (так як стіна внутрішня);
    - v. Коефіцієнт теплового випромінювання зовнішньою поверхнею,  $\varepsilon$  – 0 (так як стіна внутрішня);
    - vi. Коефіцієнти теплообміну внутрішньої і зовнішньої поверхонь огорожувальної конструкції – відповідно 8.7 та 12.0 відповідно (згідно таблиці 2 лабораторної роботи);
  - e. Стіни 7-8-9-10 (внутрішні):
    - i. Ширина – 7.0 метрів;

- ii. Висота – 2.7 метрів;
  - iii. Інші показники такі ж як і у стін 4-6.
3. Вікна 1-3:
- a. Ширина – 1.5 метрів;
  - b. Висота – 1.2 метрів;
  - c. Тип – склопакет з двома стеклами без покриття;
  - d. Випромінювальна здатність – 0.89;
  - e. Напрямок – північ;
  - f. Коефіцієнт теплопередачі для газового заповнення (повітря): 3.3;
  - g. Коефіцієнт пропускання сонячної енергії для випромінювання, перпендикулярного до вікна приймається рівним 0.75 згідно з даними таблиці 5.
4. Визначена температура під час опалення – 20 °С;
5. Визначена температура під час охолодження – 25 °С;
6. Локація:
- a. Назва міста: Кропивницький.
  - b. Середня температура відносно місяця: згідно таблиці «А.2 Кліматичні дані», ДСТУ 9090:2022.
  - c. Тривалість місяців: згідно таблиці «А.1 Щомісячна тривалість часових інтервалів», ДСТУ 9090:2022.
  - d. Температура на вулиці: згідно таблиці «А.3 Погодинна температура зовнішнього повітря репрезентативного дня місяця» , ДСТУ 9090:2022.
  - e. Середньомісячні дози сумарної сонячної радіації, осередненої для однієї години, що надходять на горизонтальну та вертикальну поверхні різного орієнтування за середніх умов хмарності – згідно таблиці А.4, ДСТУ 9090:2022.
7. Теплоізоляційна установка:
- a. Чи використовується – так або ні, залежно від режиму.
  - b. Назва: Роторний теплообмінник.
  - c. Ефективність утилізування теплоти – 0.7.
  - d. Частка повітряного потоку, яка проходить через установку – 0.5.

Таблиця з даними назв та теплопровідностей непрозорих матеріалів, які використовуються для розрахунку моделі:

Назва матеріалу	Теплопровідність $\lambda_p$ , Вт/(м · К)
Газо- та пінозобетон	0.5
Штукатурка вапняна темно-сіра	0.81
Цегла керамічна	0.64

Відповідно, для оптимального сценарію запускається теплова установка, беруться показники з наданих таблиць для склопакета з двох вікон та покриттям.

## Результати:

```
MONTH: January
- Total Energy Demand | Heating: 1204
MONTH: February
- Total Energy Demand | Heating: 1034
MONTH: March
- Total Energy Demand | Heating: 805
MONTH: April
- Total Energy Demand | Heating: 506
MONTH: May
- Total Energy Demand | Cooling: 105
MONTH: June
- Total Energy Demand | Cooling: 306
MONTH: July
- Total Energy Demand | Cooling: 407
MONTH: August
- Total Energy Demand | Cooling: 358
MONTH: September
- Total Energy Demand | Heating: 203
MONTH: October
- Total Energy Demand | Heating: 605
MONTH: November
- Total Energy Demand | Heating: 906
MONTH: December
- Total Energy Demand | Heating: 1103
```

Після встановлення теплообміннику (ротаторний теплообмінник):

```
MONTH: January
- Total Energy Demand | Heating: 1052
MONTH: February
- Total Energy Demand | Heating: 855
MONTH: March
- Total Energy Demand | Heating: 687
MONTH: April
- Total Energy Demand | Heating: 439
MONTH: May
- Total Energy Demand | Cooling: 86
MONTH: June
- Total Energy Demand | Cooling: 258
MONTH: July
- Total Energy Demand | Cooling: 353
MONTH: August
- Total Energy Demand | Cooling: 305
MONTH: September
- Total Energy Demand | Heating: 185
MONTH: October
- Total Energy Demand | Heating: 501
MONTH: November
- Total Energy Demand | Heating: 804
MONTH: December
- Total Energy Demand | Heating: 958
```

Можна зробити висновок, що споживання дещо знизилось.

### Лістинг:

*main.rs* – Точка входу в програму.

```
use crate::model::Mode;

pub mod building;
pub mod context;
pub mod date;
pub mod error;
pub mod heat_recovery_plant;
pub mod json;
pub mod location;
pub mod model;

pub const JSON_FILE_NAME: &str = "data.json";

fn main() {
    let context = match json::parse(JSON_FILE_NAME) {
        Ok(value) => value,
        Err(err) => {
            println!("Error: {}", err);
            std::process::exit(1);
        },
    };

    for (index, month) in date::MONTHS.iter().enumerate() {
        println!("MONTH: {month}");

        let total_energy_demand_heating =
            model::energy_demand(Mode::Heating, (index + 1) as u32, &context);
        let total_energy_demand_cooling =
            model::energy_demand(Mode::Cooling, (index + 1) as u32, &context);

        let diff = total_energy_demand_heating - total_energy_demand_cooling;
        let mode = if diff >= 0.0 { "Heating" } else { "Cooling" };

        println!("- Total Energy Demand | {}: {}", mode, f32::abs(diff));

        println!();
    }
}
```

*building.rs* – Додаткові структури для десеріалізації даних.

```
use serde::Deserialize;

#[derive(Deserialize, Debug)]
pub struct Building {
    pub apartment_area: f32,
    pub walls: Vec<Wall>,
    pub windows: Vec<Window>,
}

#[derive(Deserialize, Debug)]
pub struct Wall {
    pub width: f32,
    pub height: f32,
```

```

    pub layers: Vec<WallLayer>,

    pub absorption_coefficient: f32,
    pub emissivity_coefficient: f32,

    pub inner_surface_heat_transfer_coefficient: f32,
    pub outer_surface_heat_transfer_coefficient: f32,
}

#[derive(Deserialize, Debug)]
pub struct WallLayer {
    pub material: String,
    pub thickness: f32,
    pub thermal_conductivity: f32,
}

#[derive(Deserialize, Debug)]
pub struct Window {
    pub width: f32,
    pub height: f32,

    pub panes_amount: f32,

    pub glass_type: String,

    pub emissivity_coefficient: f32,
    pub heat_transfer_coefficient: f32,
    pub solar_transmittance_coefficient: f32,

    pub orientation: WindowOrientation,
}

#[derive(Deserialize, Debug)]
pub enum WindowOrientation {
    North,
    South,
    West,
    East,
}

```

### *context.rs* – Контекст застосунку.

```

use crate::building::Building;
use crate::heat_recovery_plant::HeatRecoveryPlant;
use crate::location::Location;
use serde::Deserialize;

#[derive(Deserialize, Debug)]
pub struct Context {
    pub building: Building,

    pub location: Location,

    // θint,H,set
    pub indoor_temp_heating: f32,

    // θint,C,set
    pub indoor_temp_cooling: f32,

    pub heat_recovery_plant: HeatRecoveryPlant,
}

```

*date.rs* – Константи, які стосуються дат.

```
pub const DAYS_IN_MONTHS: &[u32] = &[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31];
pub const HOURS_IN_MONTHS: &[u32] = &[744, 672, 744, 720, 744, 720, 744, 744,
720, 744, 720, 744];
pub const MONTHS: &[&str] = &[
    "January",
    "February",
    "March",
    "April",
    "May",
    "June",
    "July",
    "August",
    "September",
    "October",
    "November",
    "December",
];
```

*error.rs* – Словник помилок.

```
use thiserror::Error;

#[derive(Error, Debug)]
pub enum LabError {
    #[error("Parsing JSON data failed.")]
    ParseFailed(String),

    #[error("Reading JSON data failed.")]
    ReadFailed(String),
}
```

*heat\_recovery\_plant.rs* – Структура, яка стосується додаткової установки для теплоізоляції.

```
use serde::Deserialize;

#[derive(Deserialize, Debug)]
pub struct HeatRecoveryPlant {
    pub is_enabled: bool,

    pub title: String,
    pub wind_flow_part: f32,
    pub efficiency: f32,
}
```

*json.rs* – Десеріалізація даних.

```
use crate::context::Context;
use crate::error::LabError;
use std::fs;

pub fn parse(file_name: &str) -> Result<Context, LabError> {
    let json_content =
        fs::read_to_string(file_name).map_err(|err|
LabError::ReadFailed(err.to_string()))?;

    serde_json::from_str(&json_content).map_err(|err|
LabError::ParseFailed(err.to_string()))
```

```
}
```

*location.rs* – Структура, в якій зберігаються всі дані, пов’язані з певною локацією.

```
use serde::Deserialize;

#[derive(Deserialize, Debug)]
pub struct Location {
    pub city: String,

    pub avg_temperature_per_month: Vec<f32>,
    pub solar_radiation_per_month: Vec<f32>,

    pub outdoor_temperature: Vec<Vec<f32>>,
}
```

*model.rs* – Модель системи.

```
use crate::building::Wall;
use crate::context::Context;
use crate::date;

const EFFICIENCY_FACTOR: f32 = 1.0; // Nu-gn

// Qnd
pub fn energy_demand(mode: Mode, month: u32, context: &Context) -> f32 {
    match mode {
        Mode::Heating => {
            let total_heat_gains = total_heat_gains(month, context);
            let vent_demand = vent_pre_heat_demand(month, context);

            // Qht
            let total_heat_transfer =
                transmission_heat_transfer(&mode, month, context) + vent_demand;

            total_heat_transfer - EFFICIENCY_FACTOR * total_heat_gains -
vent_demand
        },
        Mode::Cooling => {
            let total_heat_gains = total_heat_gains(month, context);
            let vent_demand = vent_pre_cool_demand(month, context);

            // Qht
            let total_heat_transfer =
                transmission_heat_transfer(&mode, month, context) + vent_demand;

            total_heat_transfer - EFFICIENCY_FACTOR * total_heat_gains -
vent_demand
        },
    }
}

// Qgn
fn total_heat_gains(month: u32, context: &Context) -> f32 {
    internal_heat_gains(month, context) * solar_heat_gains(month, context)
}

// Qint
fn internal_heat_gains(month: u32, context: &Context) -> f32 {
    let zone = context.building.apartment_area / 3.0;
    let sum = zone * 1.8 + zone * 2.0 + zone * 2.0;
}
```



```

    0.667 * date::DAYS_IN_MONTHS[(month - 1) as usize] as f32 * sum
}

// Qsol
fn solar_heat_gains(month: u32, context: &Context) -> f32 {
    let hours = date::HOURS_IN_MONTHS[month as usize - 1];
    let mut sum = 0.0;

    for wall in &context.building.walls {
        let insolation = 0.043
            * wall.absorption_coefficient
            * wall.emissivity_coefficient
            * wall.width
            * wall.height;
        let radiation = context.location.solar_radiation_per_month[month as
        usize - 1];
        let form_coefficient = 0.5;
        let additional_heat_flow = 2.365
            * wall.emissivity_coefficient
            * wall.absorption_coefficient
            * wall.width
            * wall.height;

        let avg_heat_flow = insolation * radiation - form_coefficient *
        additional_heat_flow;
        sum += avg_heat_flow;
    }

    for window in &context.building.windows {
        let insolation =
            0.63 * window.solar_transmittance_coefficient * window.width *
        window.height;
        let radiation = context.location.solar_radiation_per_month[month as
        usize - 1];

        let avg_heat_flow = insolation * radiation;
        sum += avg_heat_flow;
    }

    hours as f32 * sum
}

// Qve,pre-heat
fn vent_pre_heat_demand(month: u32, context: &Context) -> f32 {
    let n = date::DAYS_IN_MONTHS[month as usize - 1];
    let heat_vent_transmission = heat_vent_transmission(context);

    let mut sum = 0.0;
    for i in 1..=24 {
        let system_working = 1.0;
        let outdoor_temp = context.location.outdoor_temperature[i - 1][month as
        usize - 1];

        sum +=
            system_working * heat_vent_transmission *
        (context.indoor_temp_heating - outdoor_temp);
    }

    n as f32 * sum
}

// Qve,pre-cool
fn vent_pre_cool_demand(month: u32, context: &Context) -> f32 {
    let n = date::DAYS_IN_MONTHS[month as usize - 1];

```

```

    let heat_vent_transmission = heat_vent_transmission(context);

    let mut sum = 0.0;
    for i in 1..=24 {
        let system_working = 1.0;
        let outdoor_temp = context.location.outdoor_temperature[i - 1][month as
usize - 1];

        sum +=
            system_working * heat_vent_transmission * (outdoor_temp -
context.indoor_temp_cooling);
    }

    n as f32 * sum
}

fn heat_vent_transmission(context: &Context) -> f32 {
    let temperature_correction_factor = 1.0
        - context.heat_recovery_plant.wind_flow_part *
context.heat_recovery_plant.wind_flow_part;

    let air_waste = 112.0 / 168.0 * 0.6;

    0.336 * air_waste * temperature_correction_factor
}

// Qtr
fn transmission_heat_transfer(mode: &Mode, month: u32, context: &Context) -> f32
{
    let overall_transmission_coefficient =
overall_transmission_coefficient(context);
    let hours_in_month = date::HOURS_IN_MONTHS[month as usize - 1];
    let avg_month_temperature = context.location.avg_temperature_per_month[month
as usize - 1];

    let desired_temperature = match mode {
        Mode::Heating => context.indoor_temp_heating,
        Mode::Cooling => context.indoor_temp_cooling,
    };

    overall_transmission_coefficient
        * (desired_temperature - avg_month_temperature)
        * hours_in_month as f32
}

// Htr,adj
fn overall_transmission_coefficient(context: &Context) -> f32 {
    let mut sum = 0.0;

    for window in &context.building.windows {
        let area = window.height * window.width;
        let heat_transfer_coefficient = window.heat_transfer_coefficient;

        sum += area * heat_transfer_coefficient;
    }

    for wall in &context.building.walls {
        let area = wall.height * wall.width;
        let heat_transfer_coefficient = 1.0 / reduced_thermal_resistance(wall);

        sum += area * heat_transfer_coefficient;
    }

    sum
}

```

```

}

fn reduced_thermal_resistance(wall: &Wall) -> f32 {
    let area = wall.height * wall.width;

    area / (area * thermal_resistance_i(wall))
}

fn thermal_resistance_i(wall: &Wall) -> f32 {
    1.0 / wall.inner_surface_heat_transfer_coefficient
    + wall
        .layers
        .iter()
        .map(|layer| layer.thickness / layer.thermal_conductivity)
        .sum::<f32>()
    + 1.0 / wall.outer_surface_heat_transfer_coefficient
}

pub enum Mode {
    Heating,
    Cooling,
}

```

У системі реалізовано зчитування даних з json файлу. Його зміст:

```

{
  "building": {
    "apartment_area": 56.0,

    "walls": [
      {
        "width": 5.0,
        "height": 2.7,
        "layers": [
          {
            "material": "Цегла керамічна",
            "thickness": 0.25,
            "thermal_conductivity": 0.64
          },
          {
            "material": "Газо- та пінозолобетон",
            "thickness": 0.25,
            "thermal_conductivity": 0.5
          },
          {
            "material": "Штукатурка вапняна темно-сіра",
            "thickness": 0.02,
            "thermal_conductivity": 0.81
          }
        ]
      },

      "absorption_coefficient": 0.7,
      "emissivity_coefficient": 0.93,

      "inner_surface_heat_transfer_coefficient": 8.7,
      "outer_surface_heat_transfer_coefficient": 23.0
    ],
    {
      "width": 4.0,
      "height": 2.7,
      "layers": [

```

```

    {
      "material": "Цегла керамічна",
      "thickness": 0.25,
      "thermal_conductivity": 0.64
    },
    {
      "material": "Газо- та пінозолобетон",
      "thickness": 0.25,
      "thermal_conductivity": 0.5
    },
    {
      "material": "Штукатурка вапняна темно-сіра",
      "thickness": 0.02,
      "thermal_conductivity": 0.81
    }
  ],
  "absorption_coefficient": 0.7,
  "emissivity_coefficient": 0.93,
  "inner_surface_heat_transfer_coefficient": 8.7,
  "outer_surface_heat_transfer_coefficient": 23.0
},
{
  "width": 3.5,
  "height": 2.7,
  "layers": [
    {
      "material": "Цегла керамічна",
      "thickness": 0.25,
      "thermal_conductivity": 0.64
    },
    {
      "material": "Газо- та пінозолобетон",
      "thickness": 0.25,
      "thermal_conductivity": 0.5
    },
    {
      "material": "Штукатурка вапняна темно-сіра",
      "thickness": 0.02,
      "thermal_conductivity": 0.81
    }
  ],
  "absorption_coefficient": 0.7,
  "emissivity_coefficient": 0.93,
  "inner_surface_heat_transfer_coefficient": 8.7,
  "outer_surface_heat_transfer_coefficient": 23.0
},
{
  "width": 5.0,
  "height": 2.7,
  "layers": [
    {
      "material": "Газо- та пінозолобетон",
      "thickness": 0.25,
      "thermal_conductivity": 0.5
    }
  ],
  "absorption_coefficient": 0.0,
  "emissivity_coefficient": 0.0,

```

```

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
  },

  {
    "width": 5.0,
    "height": 2.7,
    "layers": [
      {
        "material": "Газо- та пінозолобетон",
        "thickness": 0.25,
        "thermal_conductivity": 0.5
      }
    ],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
  },

  {
    "width": 5.0,
    "height": 2.7,
    "layers": [
      {
        "material": "Газо- та пінозолобетон",
        "thickness": 0.25,
        "thermal_conductivity": 0.5
      }
    ],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
  },

  {
    "width": 7.0,
    "height": 2.7,
    "layers": [
      {
        "material": "Газо- та пінозолобетон",
        "thickness": 0.25,
        "thermal_conductivity": 0.5
      }
    ],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
  },

  {
    "width": 7.0,
    "height": 2.7,
    "layers": [
      {

```

```

        "material": "Газо- та пінозолобетон",
        "thickness": 0.25,
        "thermal_conductivity": 0.5
    }
],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
},

{
    "width": 7.0,
    "height": 2.7,
    "layers": [
        {
            "material": "Газо- та пінозолобетон",
            "thickness": 0.25,
            "thermal_conductivity": 0.5
        }
    ],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
},

{
    "width": 7.0,
    "height": 2.7,
    "layers": [
        {
            "material": "Газо- та пінозолобетон",
            "thickness": 0.25,
            "thermal_conductivity": 0.5
        }
    ],

    "absorption_coefficient": 0.0,
    "emissivity_coefficient": 0.0,

    "inner_surface_heat_transfer_coefficient": 8.7,
    "outer_surface_heat_transfer_coefficient": 12.0
}
],

"windows": [
    {
        "width": 1.5,
        "height": 1.2,

        "panes_amount": 2,

        "glass_type": "Звичайне скло",

        "orientation": "North",

        "emissivity_coefficient": 0.89,
        "heat_transfer_coefficient": 3.3,
        "solar_transmittance_coefficient": 0.75
    }
]

```

```

    },
    {
      "width": 1.5,
      "height": 1.2,

      "panes_amount": 2,

      "glass_type": "Звичайне скло",

      "orientation": "North",

      "emissivity_coefficient": 0.89,
      "heat_transfer_coefficient": 3.3,
      "solar_transmittance_coefficient": 0.75
    },
    {
      "width": 1.5,
      "height": 1.2,

      "panes_amount": 2,

      "glass_type": "Звичайне скло",

      "orientation": "North",

      "emissivity_coefficient": 0.89,
      "heat_transfer_coefficient": 3.3,
      "solar_transmittance_coefficient": 0.75
    }
  ]
},

"indoor_temp_heating": 20.0,
"indoor_temp_cooling": 25.0,

"location": {
  "city": "Кропивницький",
  "avg_temperature_per_month": [-4.9, -3.9, 0.8, 9.1, 15.2, 18.6, 20.4, 19.7,
14.7, 8.2, 2.1, -2.6],
  "solar_radiation_per_month": [14, 25, 32, 40, 55, 67, 61, 44, 28, 20, 12,
9],
  "outdoor_temperature": [
    [-5.8, -4.9, -0.6, 6.7, 11.9, 15.2, 17.1, 16.7, 12.2, 6.7, 1.3, -3.3],
    [-6.3, -5.5, -1.3, 5.8, 10.9, 14.2, 16.2, 15.6, 11.2, 5.9, 0.8, -3.7],
    [-6.8, -5.9, -1.8, 5.0, 10.1, 13.5, 15.4, 14.8, 10.4, 5.1, 0.3, -4.2],
    [-7.3, -6.3, -2.3, 4.4, 9.6, 13.0, 14.9, 14.2, 9.7, 4.5, -0.1, -4.5],
    [-7.6, -6.7, -2.6, 4.0, 9.3, 12.8, 14.6, 13.8, 9.2, 4.0, -0.5, -4.8],
    [-7.9, -6.9, -2.8, 3.8, 9.3, 13.0, 14.7, 13.7, 9.0, 3.6, -0.7, -5.1],
    [-8.0, -7.0, -2.9, 3.8, 10.0, 13.7, 15.3, 14.0, 9.0, 3.4, -0.8, -5.2],
    [-8.1, -6.9, -2.6, 4.6, 11.2, 14.9, 16.5, 15.1, 9.7, 3.4, -0.9, -5.3],
    [-7.8, -6.4, -1.8, 5.9, 12.8, 16.5, 18.1, 16.7, 11.1, 4.0, -0.6, -5.1],
    [-6.9, -5.4, -0.6, 7.6, 14.7, 18.4, 19.9, 18.6, 13.0, 5.5, 0.2, -4.5],
    [-5.6, -4.2, 0.8, 9.6, 16.7, 20.2, 21.8, 20.8, 15.1, 7.4, 1.4, -3.4],
    [-4.2, -2.9, 2.2, 11.5, 18.5, 21.9, 23.6, 22.7, 17.2, 9.5, 2.8, -2.1],
    [-2.9, -1.8, 3.4, 13.1, 19.9, 23.2, 25.0, 24.3, 19.0, 11.3, 4.0, -1.0],
    [-2.0, -1.1, 4.2, 14.1, 20.8, 24.1, 25.9, 25.4, 20.1, 12.6, 4.8, -0.2],
    [-1.7, -0.8, 4.5, 14.4, 21.1, 24.4, 26.2, 25.7, 20.5, 13.0, 5.1, 0.1],
    [-1.8, -0.9, 4.4, 14.3, 21.0, 24.3, 26.1, 25.6, 20.4, 13.0, 5.0, 0.1],
    [-1.9, -1.0, 4.2, 14.0, 20.6, 23.8, 25.7, 25.2, 20.0, 12.7, 4.9, -0.1],
    [-2.2, -1.3, 3.9, 13.5, 19.9, 23.2, 25.0, 24.6, 19.5, 12.3, 4.7, -0.3],
    [-2.5, -1.7, 3.4, 12.8, 19.1, 22.3, 24.2, 23.8, 18.7, 11.8, 4.3, -0.6],
    [-3.0, -2.1, 2.9, 11.9, 18.0, 21.2, 23.1, 22.7, 17.8, 11.1, 3.9, -0.9],
    [-3.5, -2.6, 2.2, 11.0, 16.8, 20.0, 22.0, 21.6, 16.8, 10.3, 3.4, -1.4],
    [-4.0, -3.2, 1.5, 9.9, 15.5, 18.8, 20.7, 20.3, 15.6, 9.4, 2.9, -1.8],

```

```
        [-4.6, -3.7, 0.8, 8.8, 14.2, 17.5, 19.5, 19.1, 14.5, 8.5, 2.4, -2.3],  
        [-5.2, -4.3, 0.1, 7.8, 13.0, 16.3, 18.3, 17.8, 13.3, 7.6, 1.8, -2.8]  
    ]  
},  
  
    "heat_recovery_plant": {  
        "is_enabled": false,  
        "title": "None",  
        "wind_flow_part": 0.0,  
        "efficiency": 0.0  
    }  
}
```

**Висновок:** У результаті проведеної лабораторної роботи було розраховано витрати енергії на опалення та охолодження приміщення до і після проведення енергозберігаючих заходів, таких як встановлення теплоізоляції та заміна вікон.