

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 1

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 1

Виконав(ла):
Студент групи ШІ-11
Корнілов Артем Сергійович

Львів 2024

Тема роботи:

Налаштування робочого середовища (Visual Studio Code, GitHub, MSYS2, Trello, Algotester). Системи числення. Програмування на C++. Створення блок-схем.

Мета роботи:

Ознайомитись з git та його командами, принципами роботи з GitHub та створення блок-схем. Конфігурація VS Code та налаштування компілятора для C++. Використання на практиці знань з систем числення. Налаштування дошки завдань та робота в команді.

Теоретичні відомості:

Тема №1.

Git tutorial for Beginners:

<https://www.youtube.com/watch?v=RG0j5yH7evk>

Тема №2

Уроки програмування на C++:

<https://acode.com.ua/uroki-po-cpp/>

Тема №3

How To Use TRELLO for Beginners:

<https://www.youtube.com/watch?v=6drUzoeHZkg>

Тема №4

Debugging C++ Program in Visual Studio Code:

https://www.youtube.com/watch?v=2VokW_Jt0oM&t=610s

Тема №5

Просто про Системи числення: двійкова, шістнадцяткова, десяткова:

<https://www.youtube.com/watch?v=yGmVLDenVpE>

Тема №6

Арифметичні дії в різних системах числення

<https://studfile.net/preview/1582350/page:6/>

Тема №7

Складання блок-схем

Було опрацьовано на практичній.

Тема №8

Ознайомлення з Linux-командами

<https://www.freecodecamp.org/news/the-linux-commands-handbook/#heading-introduction-to-linux-and-shells>

Індивідуальний план опрацювання теорії:

Тема №1.

Ознайомився з git та його командами.

Витратив 50 хвилин.

Тема №2.

Ознайомився з базовим синтаксисом C++.

Витратив 4-5 годин.

Тема №3.

Ознайомився з Trello.

Витратив 20 хвилин.

Тема №4.

Налаштував VS Code для C++.

Витратив 3 дні.

Тема №5-6.

Ознайомився з системами числення та операціями з ними.

Витратив 20-30 хвилин.

Тема №7.

Ознайомився з інтерфейсом Draw.io та базовими блок-схемами.

Витратив 30 хвилин.

Тема №8

Ознайомився з командами Linux для їх використання в Linux терміналах.

Витратив 20 хвилин.

Виконання роботи:

1) Опрацювання завдання та вимог до програм та середовища

Завдання №1:

Обчислення складних відсотків за депозитом:

- Обчислити складні відсотки для депозиту, який був відкритий в банку на певний період часу під фіксовані відсотки з різними варіантами виплати відсотків.

Вимоги:

Використати функції *scanf* та *printf* для зчитування і форматування вводу/виводу;
В кінці програма має вивести повну інформацію про вкладені кошти, загальну суму інвестиції і суму самого заробітку.

Завдання №2:

Депутатські гроші:

Часто-густо громадяни намагаються з'ясувати, наскільки багатими є депутати. Дехто вірить, що матеріальні статки окремих депутатів є необмеженими.

Тож уявіть собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує *nn* гривень.

Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?

Вимоги:

У єдиному рядку задано одне натуральне число *n* — вартість подарунку.

$$1 \leq n \leq 10^9$$

Завдання №3

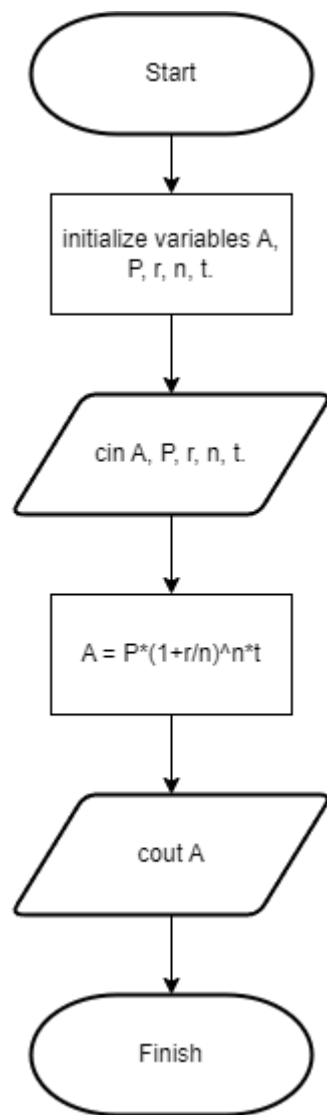
Завдання на калькуляції в двійковій системі.

- 1 Згенерувати в рандомайзері десятикове число *y* від 20 до 99
- 2 Згенерувати в рандомайзері десятикове число *x* від 20 до 99
- 3 Перевести *y* у двійкову систему числення
- 4 Перевести *x* у двійкову систему числення
- 5 Додати два двійкових числа *x* та *y*
- 6 Відняти від більшого двійкового числа менше двійкове число
- 7 Більше двійкове число поділити на менше двійкове число
- 8 Більше двійкове число помножити на менше двійкове число
- 9 Згенерувати в рандомайзері десятикове число *k* від 20 до 99
- 10 Перевести *k* у 16-ву систему числення

2) Дизайн та планована оцінка часу виконання завдань:

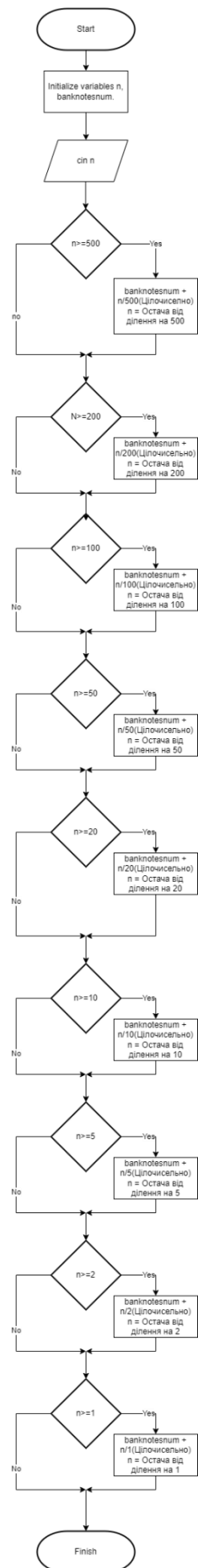
Завдання №1:

Планований час виконання: 30-35 хвилин.



Завдання №2:

Планований час виконання: 25 хвилин.

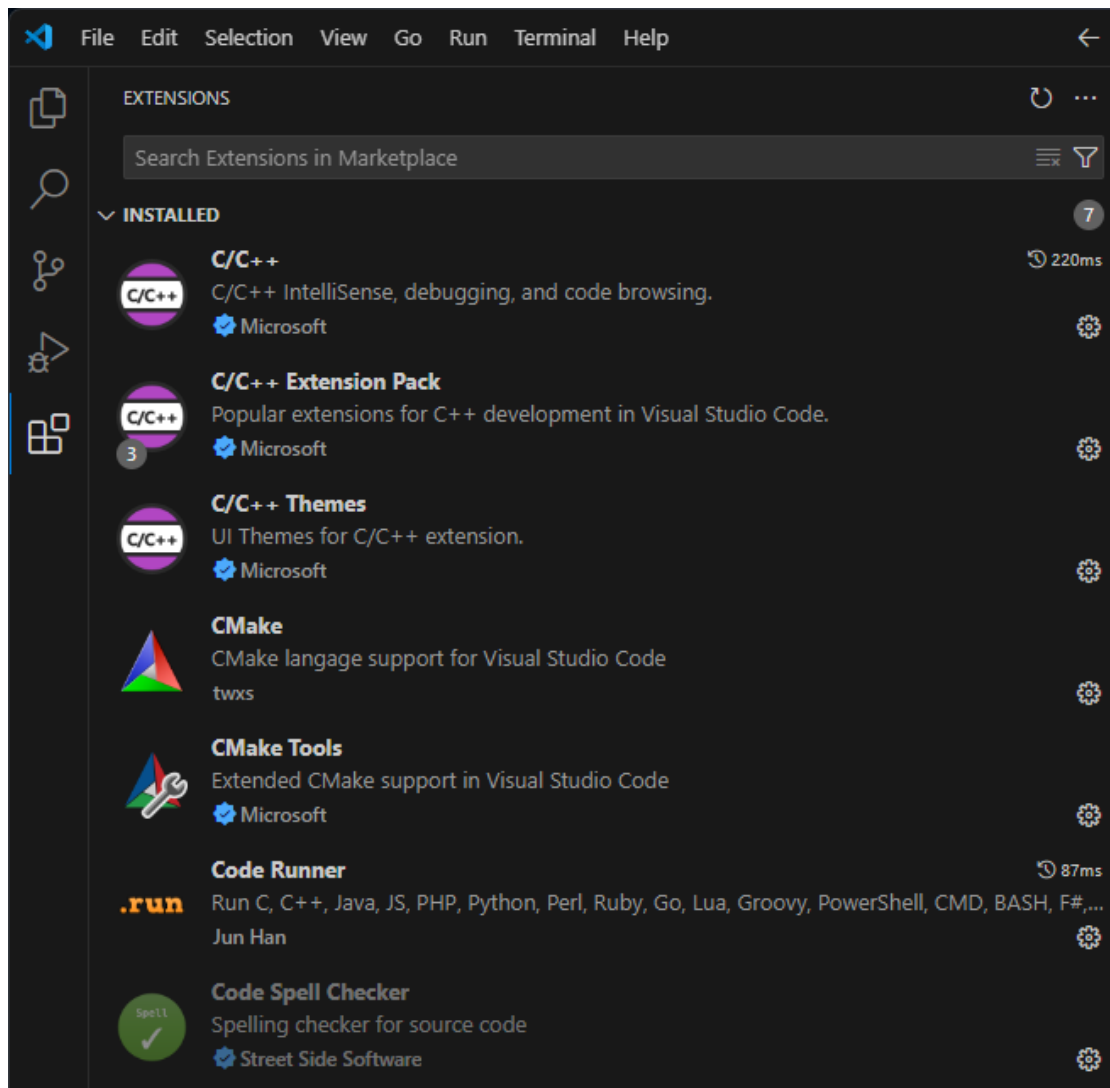


Завдання №3:

Планований час виконання: 20 хвилин.

3) Конфігурація середовища до виконання завдань:

РОЗШИРЕННЯ



КОМПІЛЯТОР

```
C:\Windows\System32>g++ --version
g++ (Rev1, Built by MSYS2 project) 14.2.0
Copyright (C) 2024 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

C:\Windows\System32>
```

Git та його базові команди

```
User@DESKTOP-T5A3JFE MINGW64 ~/ai_programming_playground_2024/ai_11/kornilov_artem/epic1
$ git config --list
user.name=Artem Kornilov
user.email=temkorn97@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/artificial-intelligence-department/ai_programming_playground_2024.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main

User@DESKTOP-T5A3JFE MINGW64 ~/ai_programming_playground_2024/ai_11/kornilov_artem/epic1
$ git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
          [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
          [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
          <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

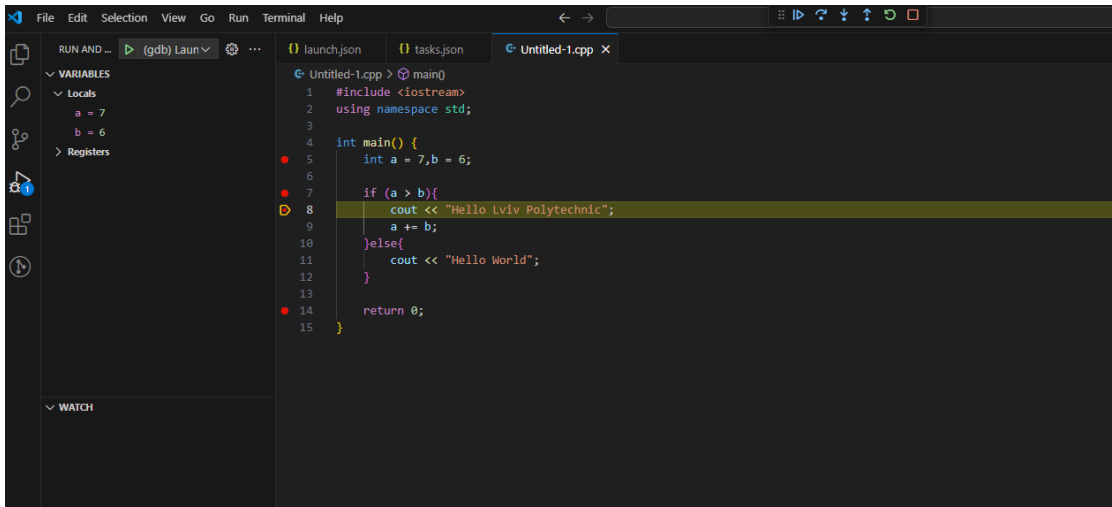
grow, mark and tweak your common history
  branch     List, create, or delete branches
  commit     Record changes to the repository
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  reset      Reset current HEAD to the specified state
  switch     Switch branches
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

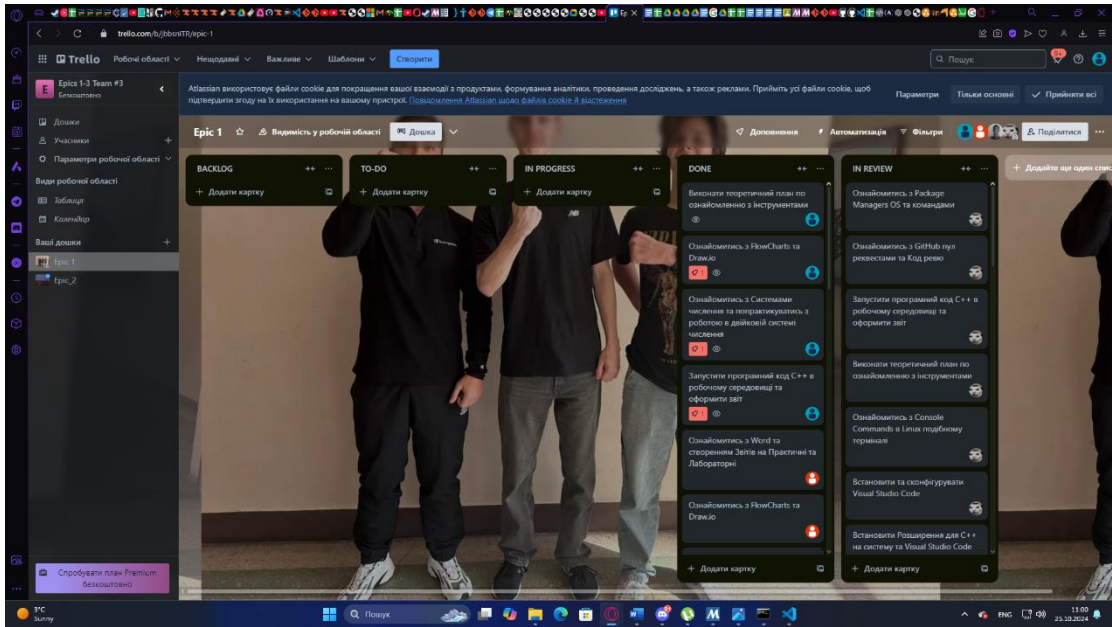
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

User@DESKTOP-T5A3JFE MINGW64 ~/ai_programming_playground_2024/ai_11/kornilov_artem/epic1
$
```

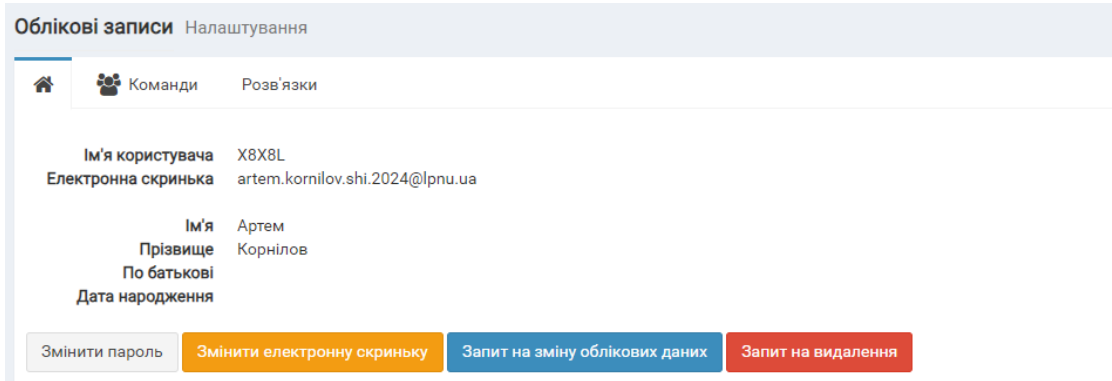

DEBUGGER



TRELLO



ALGOTESTER



4) Код програм з посиланням на зовнішні ресурси:
Завдання №1:

```
1  #include <stdio.h>
2  #include <iostream>
3  #include <cmath>
4
5  using namespace std;
6  int main() {
7      int invested;
8      float percentage;
9      float payoutsNum;
10     int yearsNum;
11     int overall;
12
13     printf ("Enter the desired amount of money to invest: \n");
14     scanf ("%d", &invested);
15
16     printf ("Enter the yearly payout percent: \n");
17     scanf ("%f", &percentage);
18
19     printf ("Enter the desired number of years you invest for: \n");
20     scanf ("%d", &yearsNum);
21
22     while (!(payoutsNum == 12) || (payoutsNum == 4) || (payoutsNum == 1)) {
23         printf ("Enter the desired number of payouts per year(either 12, 4 or 1): \n");
24         scanf ("%f", &payoutsNum);
25     }
26     float base = 1 + (percentage / payoutsNum);
27     int exponent = payoutsNum * yearsNum;
28     overall = invested * pow(base, exponent);
29     printf ("Your overall amount of money would be: %d", overall);
30
31     return 0;
32 }
```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_1_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic1/practice_work_task_1_artem_kornilov.cpp

Завдання №2:

```

1  #include <stdio.h>
2  #include <iostream>
3
4  using namespace std;
5  int main () {
6  int banknotesNum = 0;
7  int n = 0;
8
9  while (!(n >= 1) && (n <= 1000000000)) {
10 cin >> n;
11 }
12
13 if (n >= 500){
14 banknotesNum += n/500;
15 n = n % 500;
16 }
17 if (n >= 200){
18 banknotesNum += n/200;
19 n = n % 200;
20 }
21 if (n >= 100){
22 banknotesNum += n/100;
23 n = n % 100;
24 }
25 if (n >= 50){
26 banknotesNum += n/50;
27 n = n % 50;
28 }
29 if (n >= 20){
30 banknotesNum += n/20;
31 n = n % 20;
32 }
33 if (n >= 10){
34 banknotesNum += n/10;
35 n = n % 10;
36 }
37 if (n >= 5){
38 banknotesNum += n/5;
39 n = n % 5;
40 }
41 if (n >= 2){
42 banknotesNum += n/2;
43 n = n % 2;
44 }
45 if (n >= 1){
46 banknotesNum += n/1;
47 n = n % 1;
48 }

```

```

45  if (n >= 1){
46  banknotesNum += n/1;
47  n = n % 1;
48  }
49
50  cout << banknotesNum;
51
52
53  }

```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_1_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic1/self_practice_work_algotester_task_1_artem_kornilov.cpp

5) **Результат виконання завдань, тестування та фактично затрачений час:**

Завдання №1:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter the desired amount of money to invest:
10000
Enter the yearly payout percent:
0.15
Enter the desired number of years you invest for:
2
Enter the desired number of payouts per year(either 12, 4 or 1):
4
Your overall amount of money would be: 13424
```

Фактично затрачений час: 25-35 хвилин.

Завдання №2:

```
10574
25
PS C:\VS\projects> |
```

Створено	Задача	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	№
2 дні тому	0021 - Депутатські гроші	C++ 23	Зараховано	0.003	1.199	1844228

Фактично затрачений час: 30 хвилин.

Завдання №3:

$y = 12$
 $x = 41$

$y = 1100$
 $x = 101001$

1) $12 \cdot 12$
 $12 \cdot 12 = 144$

2) $41 \cdot 12$
 $41 \cdot 12 = 492$

3) $y + x =$
 $101001 + 1100 = 1101001$

$1101001_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 = 1 + 0 + 4 + 0 + 16 + 32 = 63_{10}$

$53_{10} = 46 + 7_{10}$

4) $x - y =$
 $101001 - 1100 = 101111$

$101111_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 1 + 0 + 4 + 8 + 16 = 29_{10}$

$41 - 12 = 29_{10}$

5) 101001×1100
 $101001 \times 1100 = 111101100$

$111101100_2 = 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 + 1 \cdot 2^8 = 0 + 0 + 4 + 8 + 0 + 32 + 64 + 128 + 256 = 492_{10}$

$41 \cdot 12_{10} = 492_{10}$

6) $101001 : 1100$
 $101001 : 1100 = 100$

$11_2 = 1 \cdot 2^0 + 1 \cdot 2^1 = 3_{10}$

$41_{10} : 11_{10} = 3,41(6)_{10}$

$3,41(6)_{10} \approx 3_{10}$

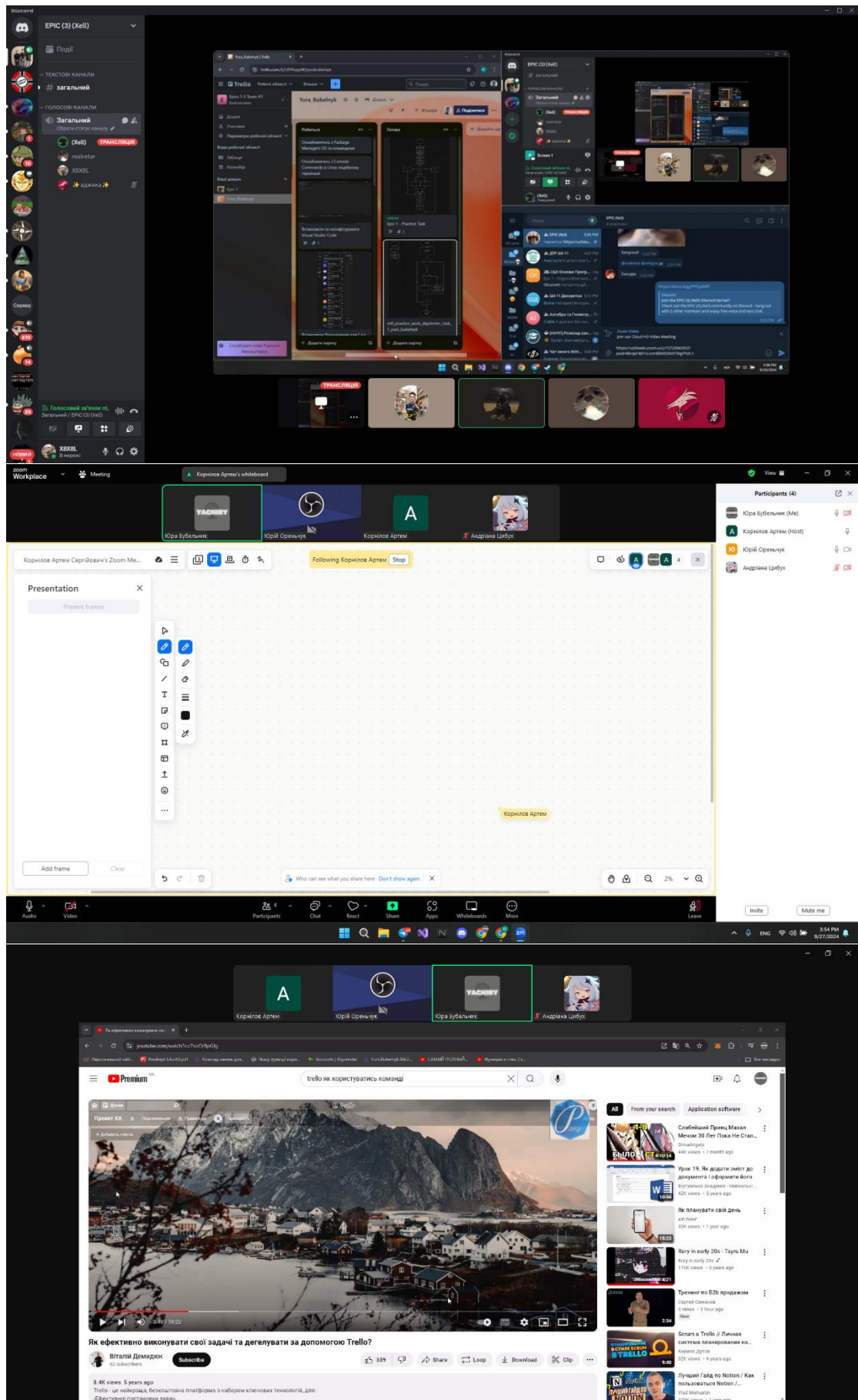
$K = 26_{10}$

7) $26 : 16$
 $26 : 16 = 1,625$

$K = 1A_{16}$

Фактично затрачений час: 20 хвилин

Робота з командою:





Висновки:

Отже, в межах цього епіку я ознайомився з Git та GitHub, налаштував середовище для написання коду, а саме Visual Studio Code, скачав плагіни необхідні для компіляції коду на C++, ознайомився з базовим синтаксисом та написав перші програми. Використовував Draw.io для складання блок-схем, що допомогло краще зрозуміти написаний код. Дізнався про системи числення та операції над ними. Створено дошку завдань в Trello для кращої роботи в команді, також були проведені зустрічі на яких ми допомагали один одному.

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/131