

**Міністерство освіти і науки України
Національний університет "Львівська Політехніка"**

Кафедра систем штучного інтелекту

Сага №1
з дисципліни
«Основи програмування»

Виконав:
студент групи ШІ-11
Гнатюк Ярослав

Львів – 2024 р.

Виконання роботи

Частина 1

Завдання №1

Назва: Practice Work 1 Variant 21

Опис: Розробити лінійний алгоритм для розв'язання задачі:

$$\alpha = \sqrt{|ax^2 \sin 2x + e^{-2x}(x+b)|}$$

$$\omega = 1/\cos^2 x^3 - x/\sqrt[3]{a^2 + b^2}, \text{ де } a=0,5; b=3,1; x=1,4.$$

Завдання №2

Назва: Practice Work 2 Variant 5

Опис: Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає порядковому номеру студента в журналі викладача

$$y = \begin{cases} \sin|ax+b|, & |x| < a/b \\ \cos|ax-b|, & |x| \geq a/b \end{cases}$$

$$\text{де } x \in [-5,5]; h_x = 1.$$

Завдання №3

Назва: Practice Work 3 Variant 15.1

Опис: Обчислення об'єму паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму паралелепіпеда.

Введіть в одному рядку довжину, ширину і висоту паралелепіпед (в сантиметрах).

Числа розділяйте пропусками.

Після введення останнього числа натискуйте <Enter> 7.5 2.5 3

Об'єм паралелепіпеда 56,25 см.куб.

Завдання №4

Назва: Practice Work 4 Variant 15.2

Опис: Обчислення вартості покупки складається з помідорів, огірків і декількох пучків кропу.

P.S. це була вся умова, тому в коді я припустив що ціна буде за шт. :)

Завдання №5

Назва: Practice Work 4 Variant 20

Опис: Скласти програму, яка виведе таблицю значень функцій $y=|x-2|+|x+1|$. Діапазон зміни аргументу від -4 до: 4, крок приросту аргументу 0,5.

Завдання №6

Назва: Algotester Practice Work 5 (Марічка і печиво)

Опис: В ході аналізу умови задачі можна зрозуміти, що потрібно від кожного введенного числа відняти 1 та вивести їх суму

Оригінал задачі:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/2>

Мій розв'язок:

<https://algotester.com/uk/ProblemSolution/Display/1907892>

Завдання №7

Назва: Algotester Practice Work 6 (Коля, Вася і теніс)

Опис: Проаналізувавши умову задачі, можна зробити висновок, що потрібно відслідковувати очки кожного гравця під час гри, перевіряти умови для завершення партії (не менше 11 очок і перевага в 2 очки), а також вивести загальний рахунок за кількістю виграних партій для кожного гравця. Крім того, якщо остання партія не завершена, потрібно вивести її поточний рахунок.

Оригінал задачі:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13>

Мій розв'язок:

<https://algotester.com/uk/ProblemSolution/Display/1907893>

Завдання №8

Назва: Algotester Practice Work 7 (В гостях у Марічки)

Опис: Проаналізувавши умову задачі, я прийшов висновку, що потрібно порахувати всю кількість кожної з букв які потрібні для складання слова, перевірити чи вдалося скласти слово капарка хоча б раз, а за тим скільки разів ще вдалося скласти слово парка

Оригінал задачі:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40922>

Мій розв'язок:

<https://algotester.com/uk/ProblemSolution/Display/1907897>

Завдання №9

Назва: Algotester Practice Work 8 (Lab 6 Variant 2)

Опис: У вас є шахова дошка розміром 8×8 і дуже багато фігур. Кожна клітинка може мати одне із таких значень:

- Пуста клітинка: O
- Пішак: P
- Тура: R
- Кінь: N
- Слон: B
- Король: K
- Королева: Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1). Далі йдуть **Q** запитів з координатами клітинки $x, y\{x, y\}x, y$. На кожен запит ви маєте вивести стрічку `sis_isi` — посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура — виведіть символ **X**. випадку, якщо клітинку не атакують — виведіть **O**. Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто, якщо між турою та клітинкою стоїть інша фігура — вважається, що тура атакує цю клітинку.

Оригінал задачі:

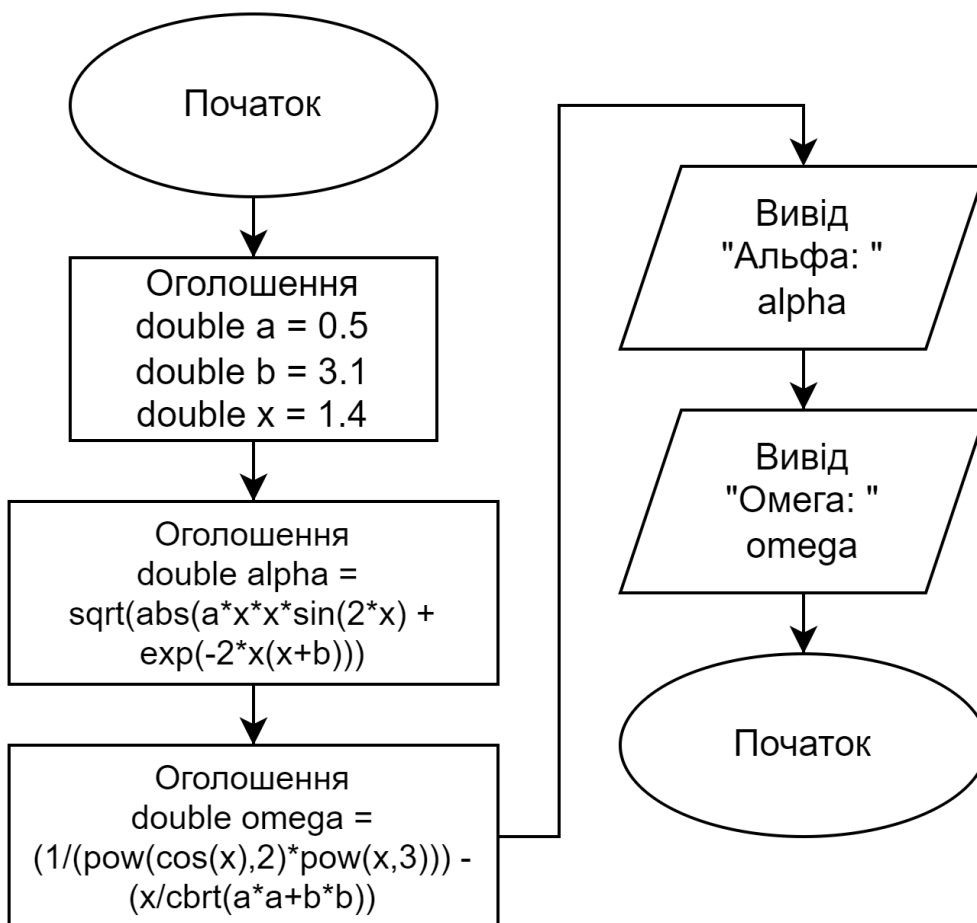
<https://algotester.com/en/ContestProblem/DisplayWithEditor/135605>

Мій розв'язок:

<https://algotester.com/en/ProblemSolution/Display/1906234>

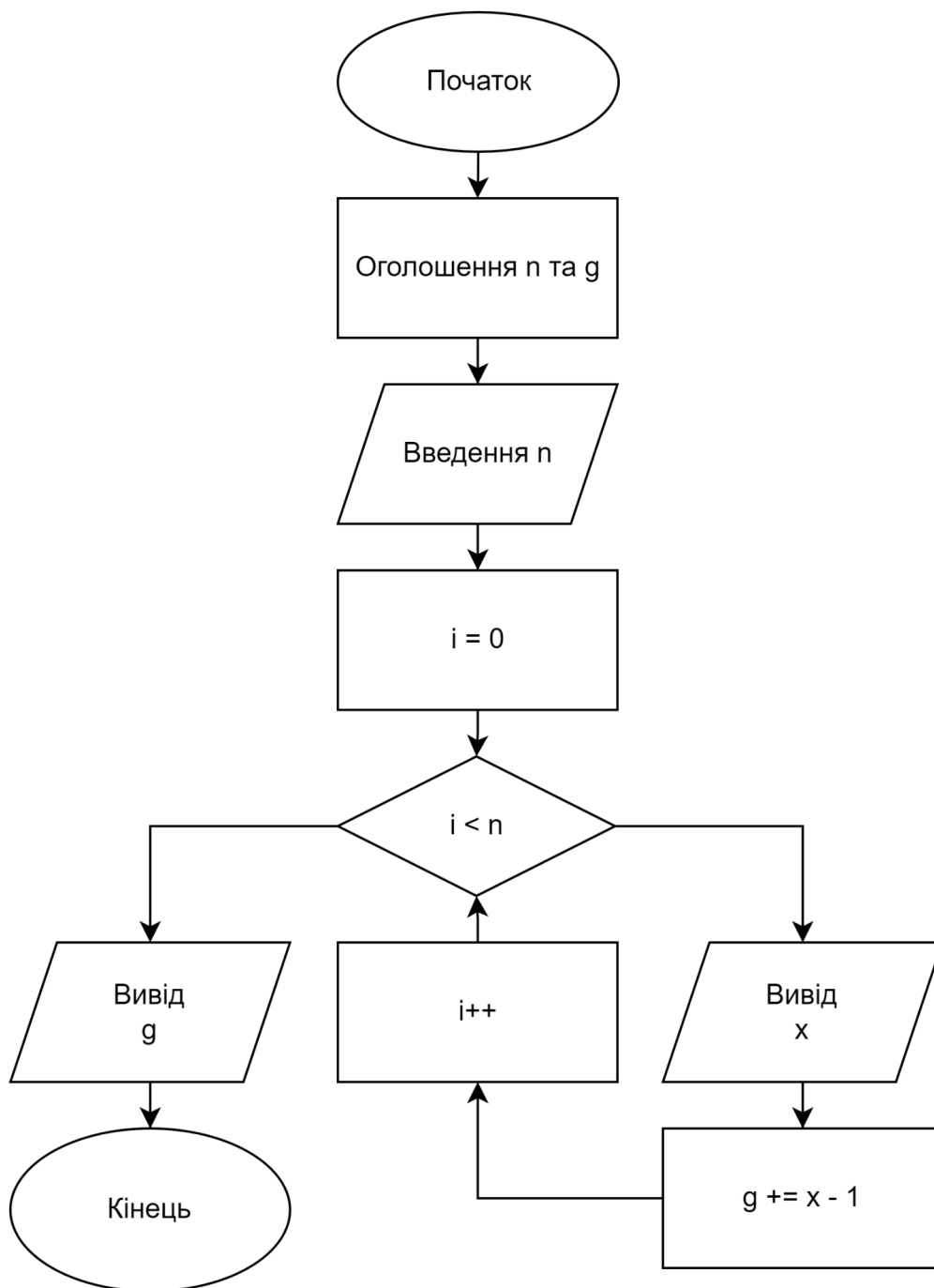
Частина 2

Завдання №1



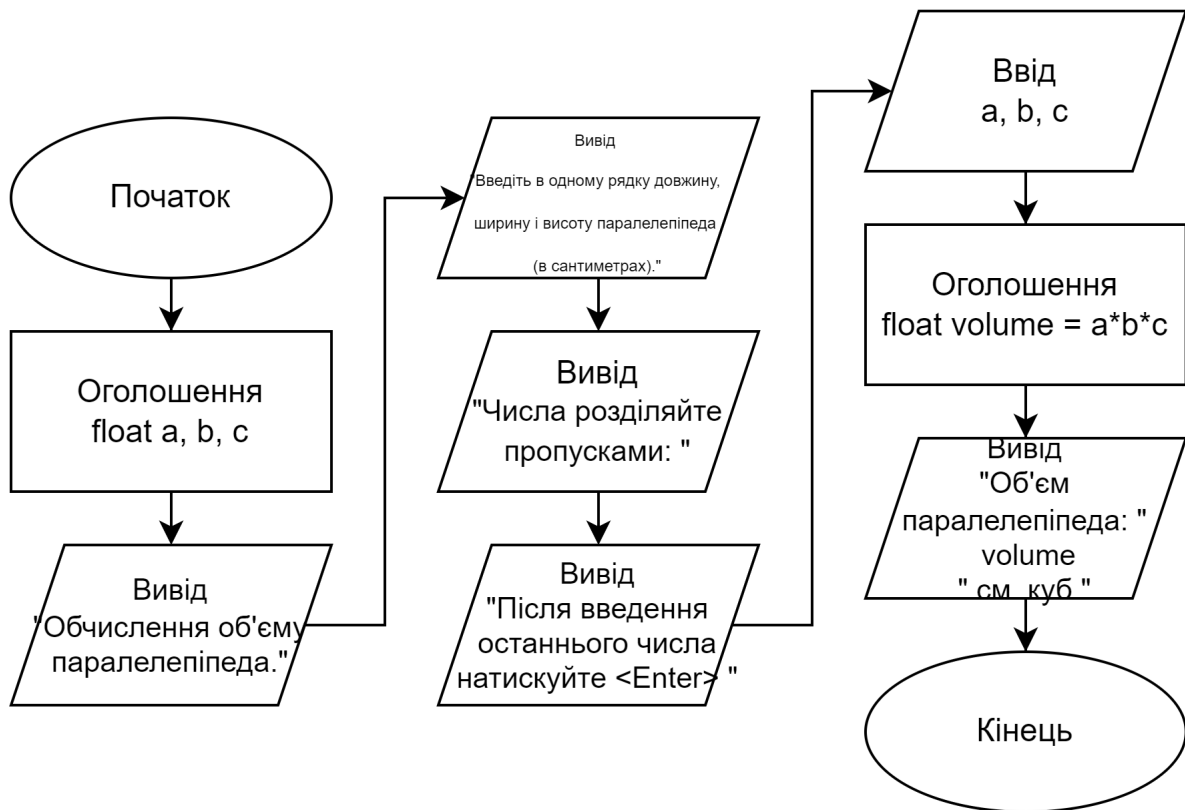
Орієнтовний час виконання: 15 хв

Завдання №2



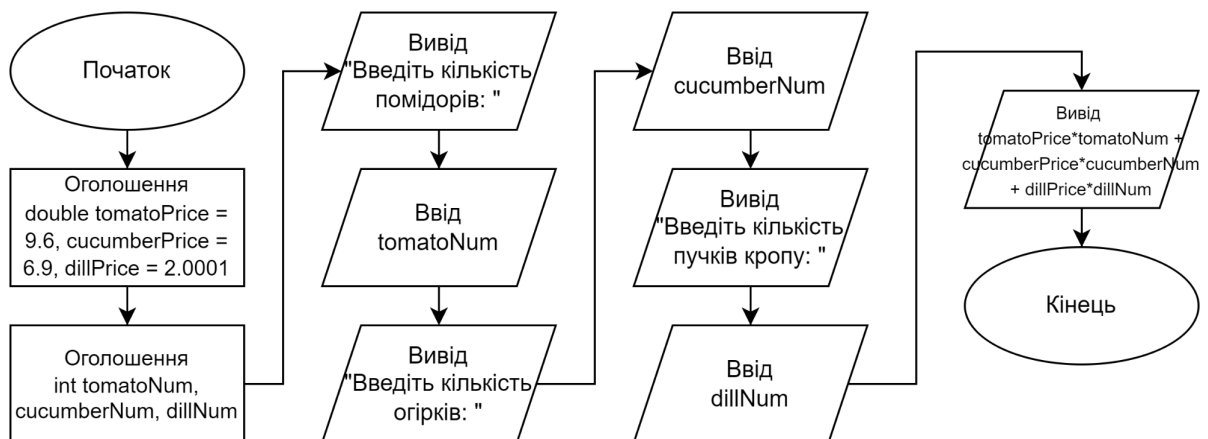
Орієнтовний час виконання: 20 хв

Завдання №3



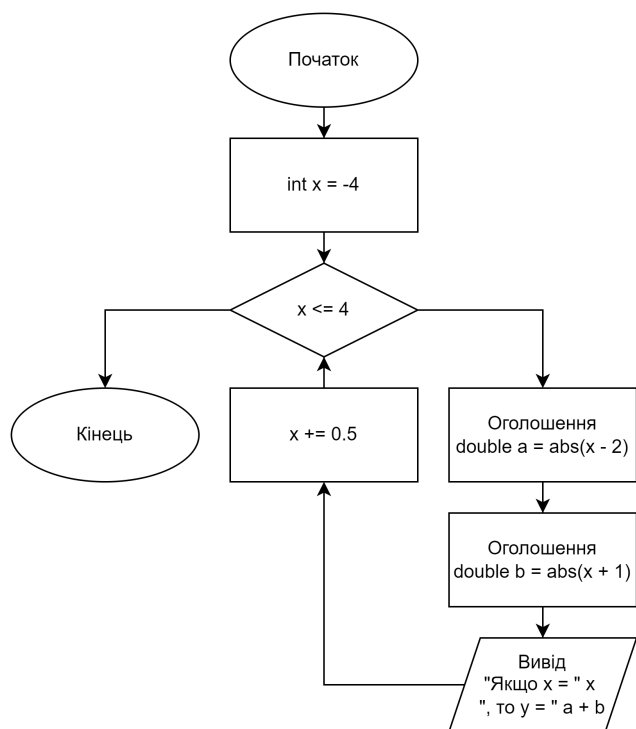
Орієнтовний час виконання: 10 хв

Завдання №4



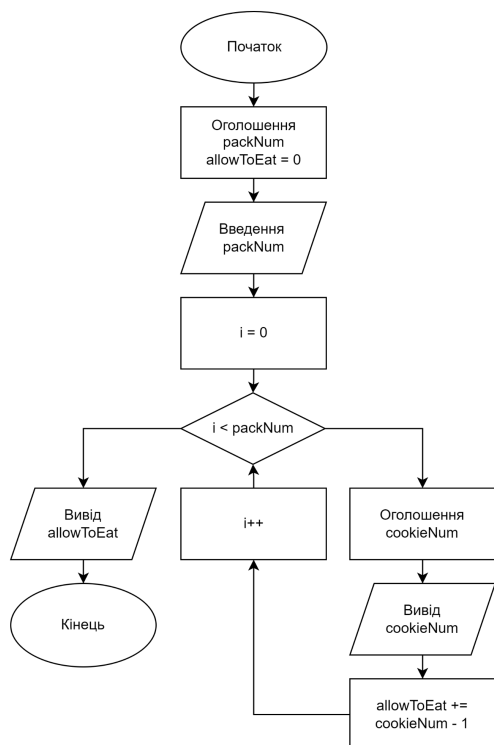
Орієнтовний час виконання: 10 хв

Завдання №5



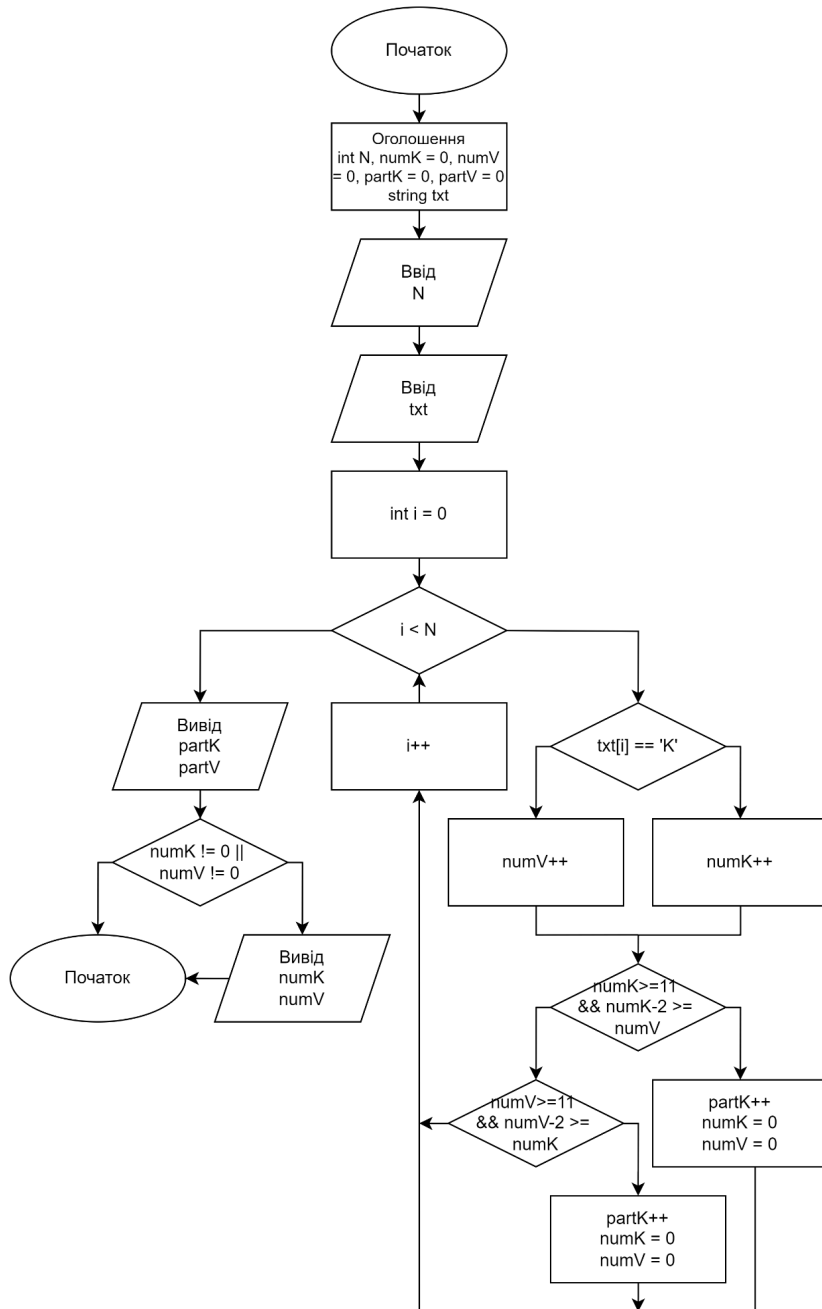
Орієнтовний час виконання: 10 хв

Завдання №6



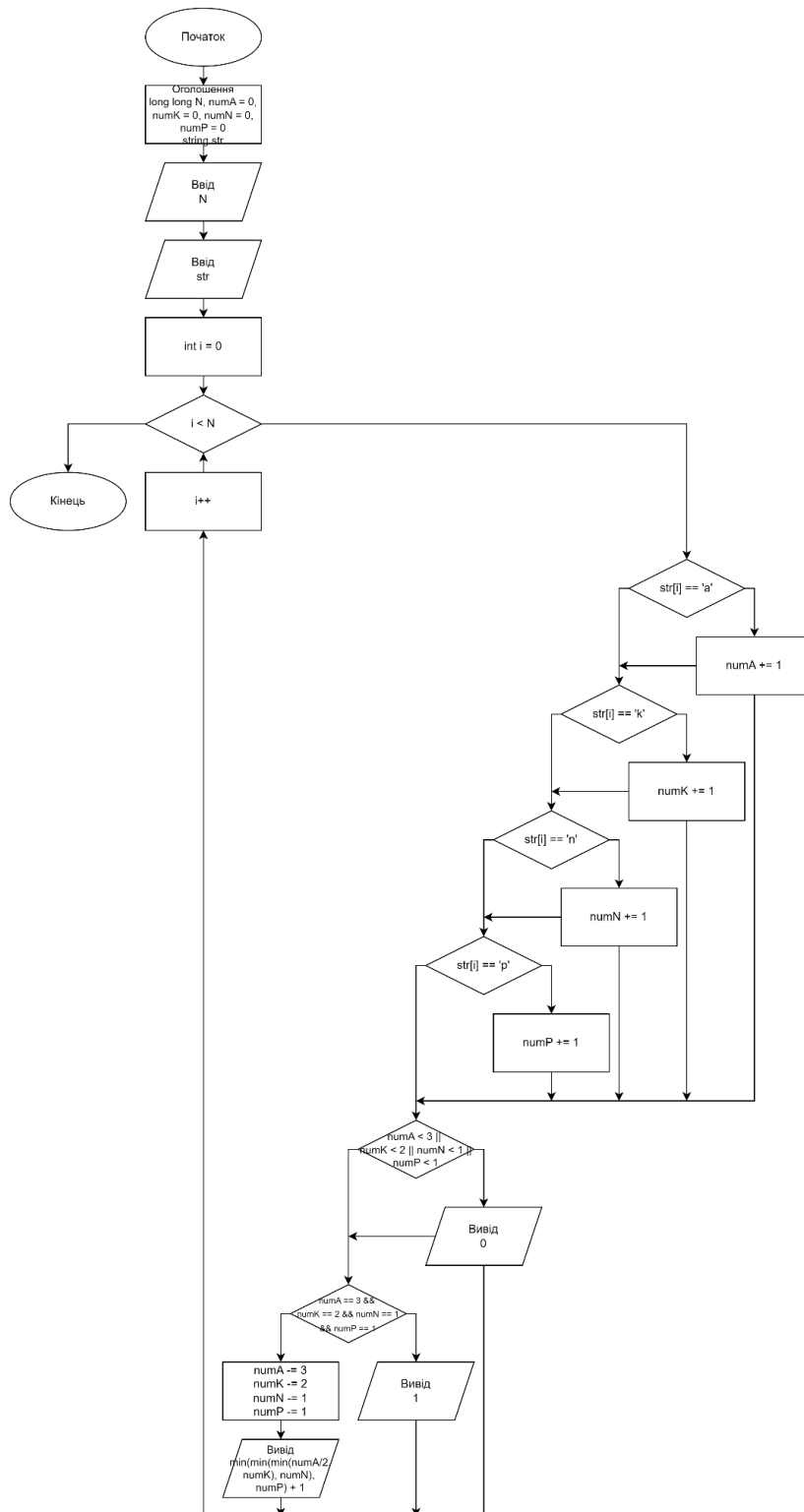
Орієнтовний час виконання: 20 хв

Завдання №7



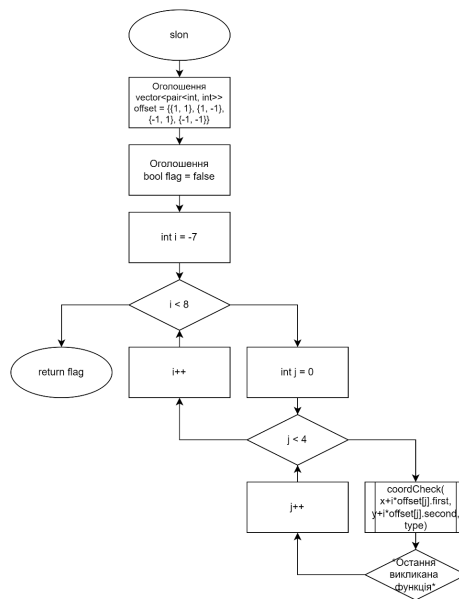
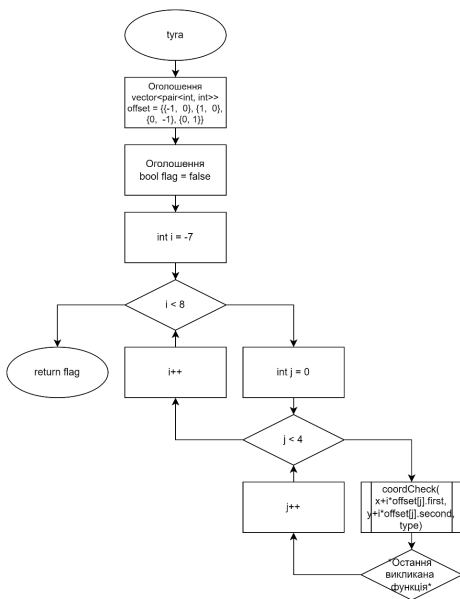
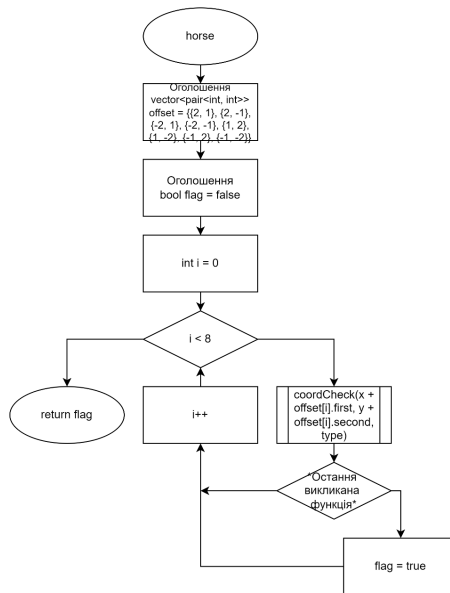
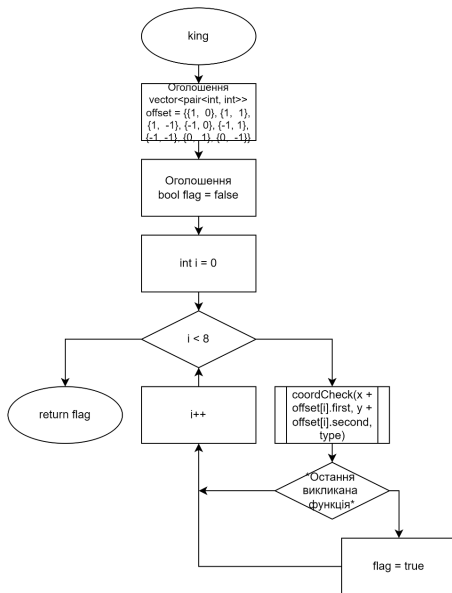
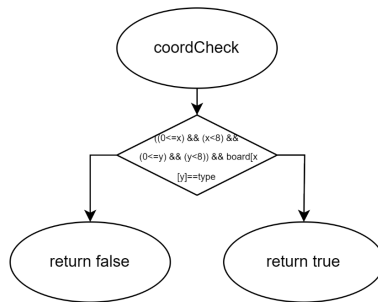
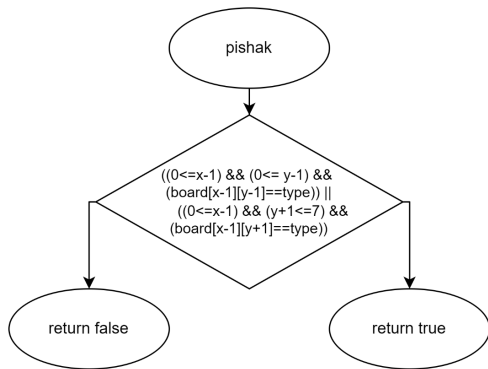
Орієнтовний час виконання: 40 хв

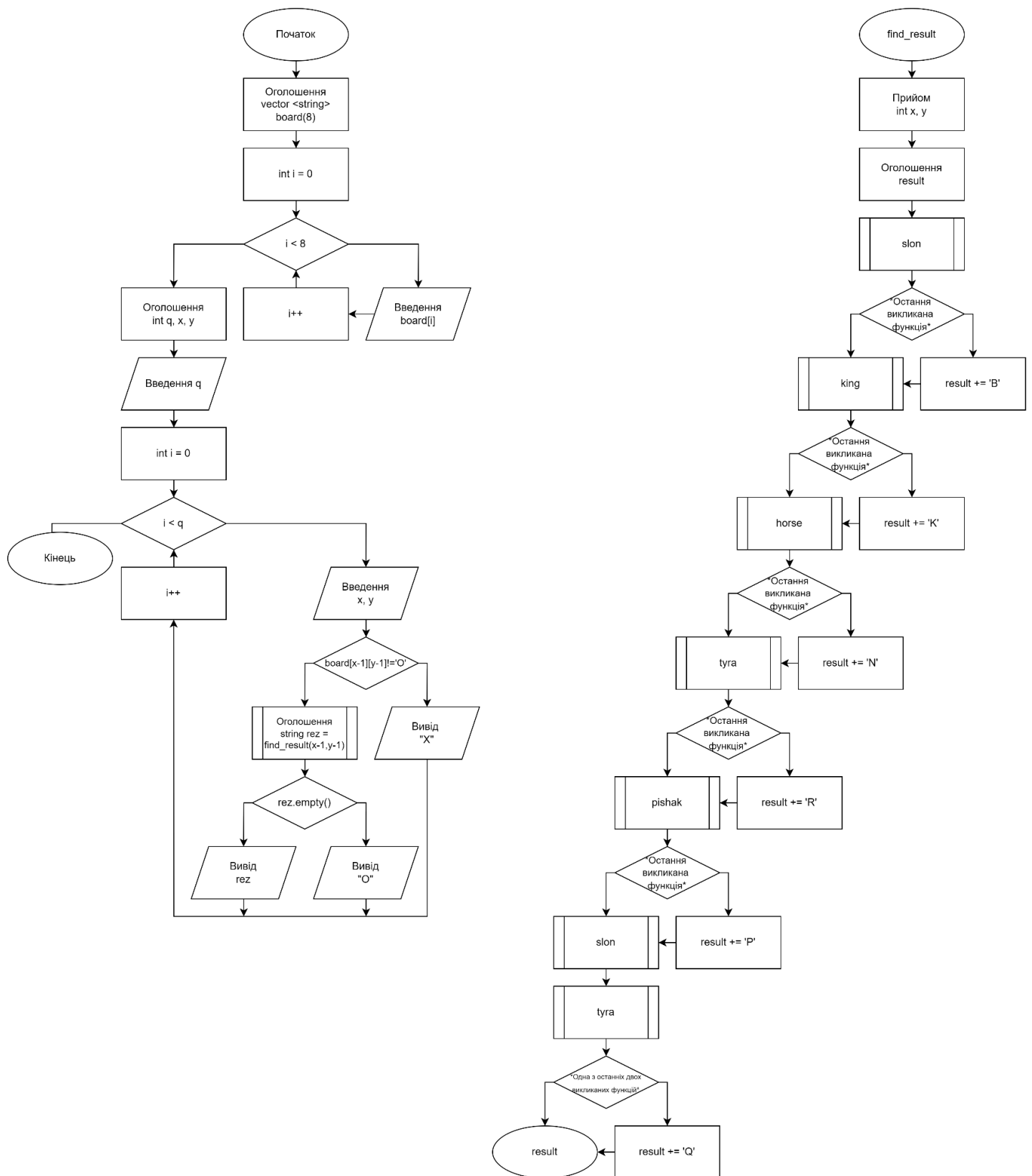
Завдання №8



Орієнтовний час виконання: 1 год

Завдання №9

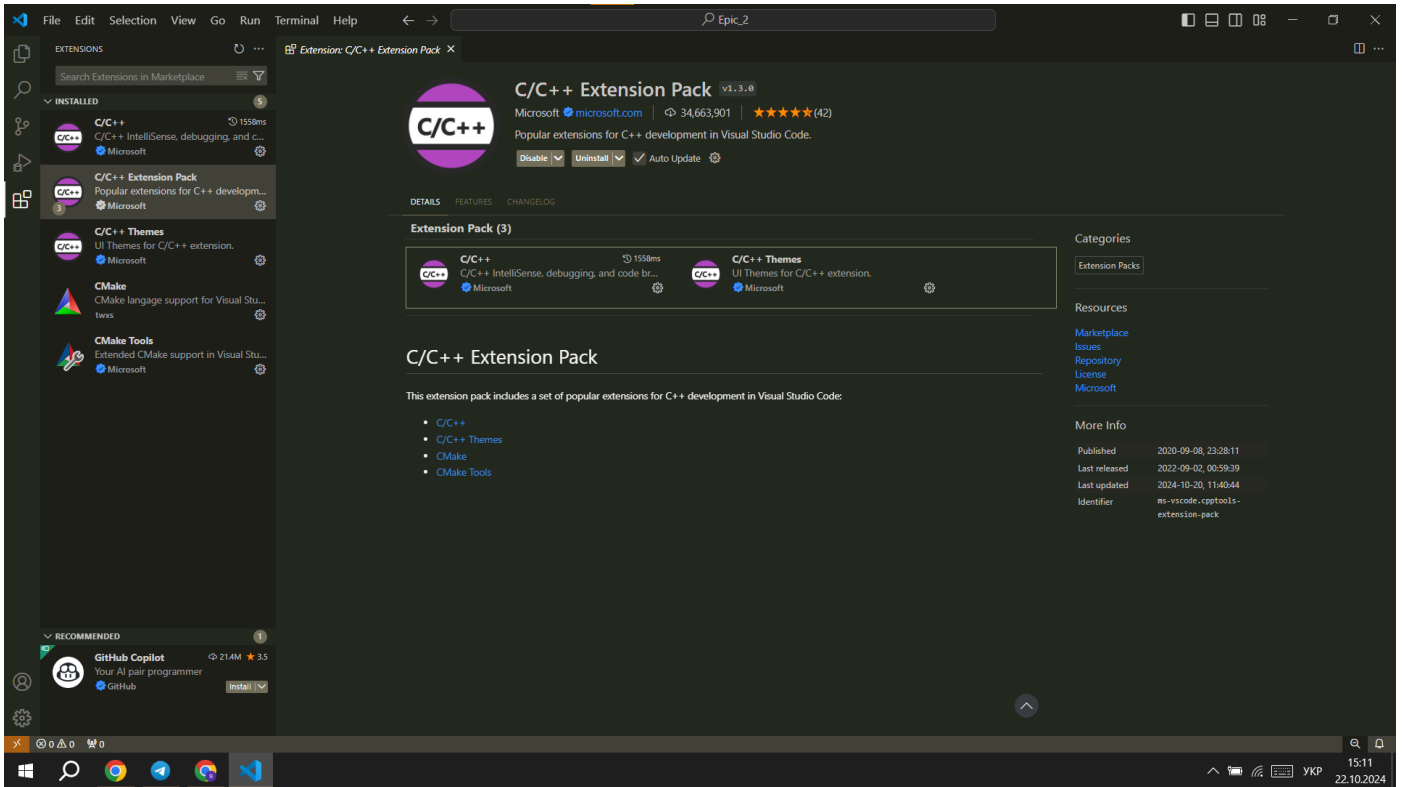




Орієнтовний час виконання: 1 год

Частина 3

Для виконання роботи використовується середовище **Visual Studio Code** зі встановленим розширенням **C/C++ Extension Pack**.



Використані бібліотеки:

- **iostream**
- **cmath**
- **vector**

Частина 4

Завдання №1

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      double a = 0.5, b = 3.1, x = 1.4;
7
8      double alpha = sqrt(abs(a * x * x * sin(2 * x) + exp(-2 * x) * (x + b)));
9      double omega = (1 / (pow(cos(x), 2) * pow(x, 3))) - (x / cbrt(a * a + b * b));
10
11     cout << "Альфа: " << alpha << endl;
12     cout << "Омега: " << omega << endl;
13
14     return 0;
15 }
```

Завдання №2

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double h = 1.0;
8
9      for (double x = -5.0; x <= 5.0; x += h) {
10         double y;
11
12         if (abs(x) <= 3) {
13             y = pow(sin(x + 0.4), 2) / (4 * x * x);
14         } else {
15             y = pow(x, 4) + 2 * x * x * cos(x);
16         }
17
18         cout << "Якщо x = " << x << ", то y = " << y << endl;
19     }
20
21     return 0;
22 }
```

Завдання №3

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      float a, b, c;
7
8      cout << "Обчислення об'єму паралелепіпеда." << endl;
9      cout << "Введіть в одному рядку довжину, ширину і висоту паралелепіпеда (в сантиметрах)." << endl;
10     cout << "Числа розділяйте пропусками: " << endl;
11     cout << "Після введення останнього числа натискуйте <Enter> ";
12
13     cin >> a >> b >> c;
14
15     float volume = a * b * c;
16
17     cout << "Об'єм паралелепіпеда: " << volume << " см. куб." << endl;
18
19     return 0;
20 }
```

Завдання №4

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      double tomatoPrice = 9.6, cucumberPrice = 6.9, dillPrice = 2.0001;
6      int tomatoNum, cucumberNum, dillNum;
7
8      cout << "Введіть кількість помідорів: ";
9      cin >> tomatoNum;
10
11     cout << "Введіть кількість огірків: ";
12     cin >> cucumberNum;
13
14     cout << "Введіть кількість пучків кропу: ";
15     cin >> dillNum;
16
17     cout << tomatoPrice * tomatoNum + cucumberPrice * cucumberNum + dillPrice * dillNum;
18
19     return 0;
20 }
```

Завдання №5

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      for (double x = -4; x <= 4; x += 0.5) {
7          double a = abs(x - 2);
8          double b = abs(x + 1);
9
10         cout << "Якщо x = " << x << ", то y = " << a + b << endl;
11     }
12
13     return 0;
14 }
```

Завдання №6

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      long long packNum, allowToEat = 0;
6      cin >> packNum;
7      for (long long i = 0; i < packNum; i++) {
8          int cookieNum;
9          cin >> cookieNum;
10         allowToEat += cookieNum - 1;
11     }
12     cout << allowToEat;
13     return 0;
14 }
```


Завдання №7

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      int N, numK = 0, numV = 0, partK = 0, partV = 0;
7      string txt;
8      cin >> N;
9      cin >> txt;
10     for (int i = 0; i < N; i++) {
11         if (txt[i] == 'K') numK++;
12         else numV++;
13
14         if (numK >= 11 && numK - 2 >= numV) {
15             partK++;
16             numK = 0;
17             numV = 0;
18         }
19         else if (numV >= 11 && numV - 2 >= numK) {
20             partV++;
21             numK = 0;
22             numV = 0;
23         }
24     }
25
26     cout << partK << ':' << partV << endl;
27     if (numK != 0 || numV != 0) cout << numK << ':' << numV;
28
29     return 0;
30 }
```

Завдання №8

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      long long N, numA = 0, numK = 0, numN = 0, numP = 0;
6      string str;
7      cin >> N;
8      cin >> str;
9      for (int i = 0; i < N; i++) {
10         if (str[i] == 'a') numA += 1;
11         else if (str[i] == 'k') numK += 1;
12         else if (str[i] == 'n') numN += 1;
13         else if (str[i] == 'p') numP += 1;
14     }
15
16     if (numA < 3 || numK < 2 || numN < 1 || numP < 1) cout << 0;
17     else if (numA == 3 && numK == 2 && numN == 1 && numP == 1) cout << 1;
18     else {
19         numA -= 3;
20         numK -= 2;
21         numN -= 1;
22         numP -= 1;
23         cout << min(min(min(numA / 2, numK), numN), numP) + 1;
24     }
25
26     return 0;
27 }
```

Завдання №9

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5  vector<string> board(8);
6
7  bool king(int x, int y, char type);
8  bool tyra(int x, int y, char type);
9  bool slon(int x, int y, char type);
10 bool horse(int x, int y, char type);
11 bool pishak(int x, int y, char type);
12 bool coordCheck(int x, int y, char type);
13 string find_result(int x, int y);
14
15 int main() {
16
17     for (int i = 0; i < 8; i++) {
18         cin >> board[i];
19     }
20
21     int q, x, y;
22     cin >> q;
23     for (int i = 0; i < q; i++) {
24         cin >> x >> y;
25         if (board[x-1][y-1] != '0') cout << 'X' << endl;
26         else {
27             string rez = find_result(x - 1, y - 1);
28             if (rez.empty()) cout << '0' << endl;
29             else cout << rez << endl;
30         }
31     }
32 }
33
34 bool coordCheck(int x, int y, char type) {
35     if ((0 <= x) && (x < 8) && (0 <= y) && (y < 8)) && board[x][y] == type)
36         return true;
37     else
38         return false;
39 }
40
```

```

41 bool king(int x, int y, char type) {
42     vector<pair<int, int>> offset = {{1, 0}, {1, 1}, {1, -1}, {-1, 0}, {-1, 1}, {-1, -1}, {0, 1}, {0, -1}};
43     bool flag = false;
44     for (int i = 0; i < 8; i++)
45         if (coordCheck(x + offset[i].first, y + offset[i].second, type))
46             flag = true;
47     return flag;
48 }
49
50 bool tyra(int x, int y, char type) {
51     vector<pair<int, int>> offset = {{-1, 0}, {1, 0}, {0, -1}, {0, 1}};
52     bool flag = false;
53     for (int i = -7; i < 8; i++) {
54         for (int j = 0; j < 4; j++) {
55             if (coordCheck(x + i * offset[j].first, y + i * offset[j].second, type))
56                 flag = true;
57         }
58     }
59     return flag;
60 }
61
62 bool slon(int x, int y, char type) {
63     vector<pair<int, int>> offset = {{1, 1}, {1, -1}, {-1, 1}, {-1, -1}};
64     bool flag = false;
65     for (int i = -7; i < 8; i++) {
66         for (int j = 0; j < 4; j++) {
67             if (coordCheck(x + i * offset[j].first, y + i * offset[j].second, type))
68                 flag = true;
69         }
70     }
71     return flag;
72 }
73
74 bool horse(int x, int y, char type) {
75     bool flag = false;
76     vector<pair<int, int>> offset = {{2, 1}, {2, -1}, {-2, 1}, {-2, -1}, {1, 2}, {1, -2}, {-1, 2}, {-1, -2}};
77     for (int i = 0; i < 8; i++) {
78         if (coordCheck(x + offset[i].first, y + offset[i].second, type))
79             flag = true;
80     }
81     return (flag);
82 }
83
84 bool pishak(int x, int y, char type) {
85     if (((0 <= x - 1) && (0 <= y - 1) && (board[x - 1][y - 1] == type)) ||
86         ((0 <= x - 1) && (y + 1 <= 7) && (board[x - 1][y + 1] == type)))
87         return true;
88     else return false;
89 }
90
91 string find_result(int x, int y) {
92     string result;
93
94     if (slon(x, y, 'B')) result += "B";
95     if (king(x, y, 'K')) result += "K";
96     if (horse(x, y, 'N')) result += "N";
97     if (tyra(x, y, 'R')) result += "R";
98     if (pishak(x, y, 'P')) result += "P";
99     if (slon(x, y, 'Q') || tyra(x, y, 'Q')) result += "Q";
100
101     return result;
102 }

```

Частина 5

Завдання №1

Альфа: 0.775844

Омега: 11.9621

Реальний час виконання: 15 хв

Завдання №2

Якщо $x = -5$, то $y = 639.183$

Якщо $x = -4$, то $y = 235.083$

Якщо $x = -3$, то $y = 0.00738171$

Якщо $x = -2$, то $y = 0.0624467$

Якщо $x = -1$, то $y = 0.0797053$

Якщо $x = 0$, то $y = \text{inf}$

Якщо $x = 1$, то $y = 0.242778$

Якщо $x = 2$, то $y = 0.0285157$

Якщо $x = 3$, то $y = 0.00181392$

Якщо $x = 4$, то $y = 235.083$

Якщо $x = 5$, то $y = 639.183$

Реальний час виконання: 25 хв

Завдання №3

Обчислення об'єму паралелепіпеда.

Введіть в одному рядку довжину, ширину і висоту паралелепіпеда (в сантиметрах).

Числа розділяйте пропусками:

Після введення останнього числа натискуйте <Enter> 3.9 4.0 2.0009

Об'єм паралелепіпеда: 31.214 см. куб.

Реальний час виконання: 15 хв

Завдання №4

Введіть кількість помідорів: 4

Введіть кількість огірків: 5

Введіть кількість пучків кропу: 5394
10861.4

Реальний час виконання: 10 хв

Завдання №5

Якщо $x = -4$, то $y = 9$

Якщо $x = -3.5$, то $y = 8$

Якщо $x = -3$, то $y = 7$

Якщо $x = -2.5$, то $y = 6$

Якщо $x = -2$, то $y = 5$

Якщо $x = -1.5$, то $y = 4$

Якщо $x = -1$, то $y = 3$

Якщо $x = -0.5$, то $y = 3$

Якщо $x = 0$, то $y = 3$

Якщо $x = 0.5$, то $y = 3$

Якщо $x = 1$, то $y = 3$

Якщо $x = 1.5$, то $y = 3$

Якщо $x = 2$, то $y = 3$

Якщо $x = 2.5$, то $y = 4$

Якщо $x = 3$, то $y = 5$

Якщо $x = 3.5$, то $y = 6$

Якщо $x = 4$, то $y = 7$

Реальний час виконання: 10 хв

Завдання №6

5

1 9 37 53 5863

5958

Реальний час виконання: 25 хв

Завдання №7

30
WVKVKKVWVVKVKKKKVWVKVKKKKVWV
1:0
2:4

Реальний час виконання: 1 год

Завдання №8

15
pkakvkсаарanank
2

Реальний час виконання: 1 год

Завдання №9

00000000
0R000000
00N00000
0000P000
00000000
00000000
K0Q00000
0000000R

7
8 1
KR
1 2
NR
5 4
NP
5 1
Q
6 2
KRQ
8 4
RQ
6 7
0

Реальний час виконання: 3 год