18 лютого на кафедрі інформатики проведено **І тур Всеукраїнської олімпіади студентів з інформатики** . В олімпіаді змагалися 13 студентів фізико-математичного факультету першого, другого та третього курсів спеціальностей інформатики та математики. За результатами змагання
перше місце
посіли
Міщик Вікторія
(група 3 ІМ) та
Юрков Євгеній
(група 2 ІМ), почесне
друге місце
зайняли
Чан Тхієн Лок
(група 3 ІА) і
Глушич Віталій
(група 3 ІМ),
третє місце
здобув студент групи 3 МФ –
Панов Олександр
. Бажаємо переможцям з гідністю захистити честь нашого університету в ІІ турі Всеукраїнської олімпіади студентів з інформатики, який відбудеться 19-21 квітня на базі Національного аерокосмічного університету імені М.Є. Жуковського «ХАІ». Пропонуємо аналіз розв'язку олімпіадних задач.

1. Дата

Допоможіть мандрівнику Тарасу, який подорожує з Лісабона до Пекіна, визначити дату його прибуття до столиці Китаю, якщо авіаперелiт триває 48 годин. Врахувати, що рік може бути високосним - його номер ділиться націло на 4 і з остачею на 100, або без остачі на 400.

Вхідні дані *– число, місяць та рік – цілі числа, вводяться з клавіатури.*

Вихідні дані *– дата, яка буде післязавтра у форматі вхідних даних, виводяться на екран* .

Приклад 1

Вхідні дані

1 8 2009

Вихідні дані

3 8 2009

Приклад 2

Вхідні дані

28 2 2016

Вихідні дані

1 3 2016

Аналіз розв’язку задачі

1) Перевіримо, чи є рік **високосним**;

2) Позначимо кількість днів у заданому місяці через **x**;

3) Збільшимо задане користувачем значення дня на два (**день = день + 2**);

4) ***Якщо*** **день > x**, ***то*** (**день = день – x**) та збільшуємо місяць на один (**місяць = місяць + 1)** ;

5) ***Якщо*** **місяць > 12, *то* місяць = місяць – 12** і збільшуємо рік на один (**рік = рік + 1**).

Тестова таблиця з результатами роботи програми має вигляд:

VhD

VuhD

Бали

1

30 12 2009

1 1 2010

3

2

28 2 2008

1 3 2008

3

3

29 9 2013

1 10 2013

3

4

30 6 2015

2 7 2015

3

5

29 10 2012

31 10 2012

3

Максимум балів

15

2. Квартира

Допоможіть Тарасу відщукати квартиру його китайського друга під номером **N**. Створіть програму
визначення номеру під'їзду
Pd
і номеру поверху
Pv
, де розташована квартира з номером
N
, якщо будинок має
m
поверхів і
k
квартир на кожному поверсі.

Вхідні дані – *N, m, k - цілі числа, вводяться з клавіатури.*

Вихідні дані *– Pd і Pv - цілі числа, виводяться на екран.*

Приклад 1

Вхідні дані

123 16 6

Вихідні дані

2 5

Приклад 1

Вхідні дані

66 6 3

Вихідні дані

4 4

Аналіз розв'язку задачі «Поштар»

Існує декілька способів розв'язання цієї популярної задачі. Проте найбільш раціональний розв'язок можна одержати, помітивши, що номер під'їзду **Pd** залежить від результату цілочисельного ділення (операція
div
) заданого номеру квартири
N
на кількість квартир у під'їзді (
k*m
), а номер поверху
Pv
- від результату цілочисельного ділення порядкового номеру квартири в під'їзді на кількість квартир на поверсі
k
.

Отже, спочатку слід визначити номер під'їзду, потім - поверху.

1. Обчислимо номер під'їзду.

На перший погляд, здається, що можна скористатися формулою:

Pd = N div (k * m)

Але уважний розгляд цієї формули показує, що

а) одержимо результат для нумерації під'їздів, починаючи з нуля, а не з одиниці. Для корекції результату вводимо у вираз доданок 1:

Pd = N div (k * m) + 1;

б) одержимо помилкові результати для останніх квартир в кожному під'їзді, номер яких є кратним (**k * m**). Для корекції результату зменшуємо на 1 номер квартири, тобто переходимо до їх нумерації, починаючи з нуля. Таким чином, остаточно формула для обчислення номеру під'їзду набуває вигляду:

Pd = (N - 1) div (k * m) + 1,

2. Обчислимо номер поверху.

Очевидно, що порядковий номер квартири **N** у під'їзді **Pd** задається виразом:

N - (k * m) * (Pd - 1).

Для обчислення номеру поверху застосовуємо дії, аналогічні п.1, а саме: додаємо до виразу 1, щоб перейти до нумерації поверхів з одиниці, і зменшуємо порядковий номер квартири на 1, щоб уникнути помилок з кожною останньою квартирою на поверсі. Отже, одержуємо таку формулу для обчислення номеру поверху:

Pv = (N - (k * m) * (Pd - 1) - 1) div k + 1

Тестова таблиця з результатами роботи програми має вигляд:

Вхідні дані

Вихідні дані

Бали

N

m

k

Pd

Pv

1

12

5

4

1

3

3

2

48

6

4

2

6

3

3

99

9

2

6

5

3

4

128

16

6

2

6

3

5

216

24

7

2

14

3

Максимум балів

15

3. Дрібна решта

Купуючи квиток на метро в Пекіні, Тарас отримав решту **m** у 2 юані та 5 мао (1 юань – це 10 мао). Він отримав одну монету в 1 юань та три монети по 5 мао. Пізніше, придбавши туристичну мапу, він знову отримав решту у розмірі 2 юані та 5 мао, яка складалася з п’яти монет по 2 мао, двох по 5 мао та пяти по 1 мао. Тарас спробував підрахувати кількість варіантів **V** набору монет для
решти в 2 юані та 5 мао і визначив, що їх 68.

Напишіть програму підрахунку кількості різних комбінацій **V** китайських монет (1 мао, 2 мао, 5 мао, 1 юань, 2 юані та 5 юанів), які можуть утворити визначену решту **m**
.

Вхідні дані *– m* **–** *ціле число в мао, вводиться з клавіатури.*

Вихідні дані *– V – кількість варіантів різних комбінацій решти, виводиться на екран.*

Приклад
Вхідні дані

17

Вихідні дані

28

Аналіз розв’язку задачі

Одним із розв’язків задачі є рекурентне співвідношення, яке можна реалізувати через ре курсію.

Створимо масив цілих чисел – **номінал_монет**, який складається з **n** елементів: 1, 2, 5, 10, 20, 50.

Функція **РЕШТА** повертає кількість способів розміну суми **m** монетами не більше індексу **n**
.

Розглянемо на прикладі рекурентне співвідношення. Маємо решту в 17 мао. Визначимо

кількість способів її видачі монетами номіналом не більше 1 юаня (індекс = 3).

Кількість способів складається з двох:

1) Видати одну монету в 1 юань і підрахувати кількість способів виплатити залишок в 7 мао монетами номіналом не більше 5 мао;

2) Не брати монети в 1 юань і підрахувати кількість способів виплатити всю суму в 17 мао монетами номіналом не більше 5 мао (індекс = 2).

Опишемо функцію **РЕШТА.**

Функція **РЕШТА** (m, n)

Початок

Якщо **m < 0** ***або*** **n < 0, то** функція повертає значення **нуль.**

Якщо **m = 0** ***або*** **n = 0, то** функція повертає значення **один.**

функція **РЕШТА** повертає значення **РЕШТА(m, n - 1) + РЕШТА(номінал_монет[n], n).**

Кінець

Викличемо функцію **РЕШТА(m, 5),** де 5 кількість елементів в масиві **номінал_монет – 1.**

Тестова таблиця з результатами роботи програми має вигляд:

m

V

Бали

1

1

1

4

2

10

11

4

3

100

4562

4

4

400

1420015

4

5

999

102613660

4

Максимум балів

20