

Управління освіти і науки
Чернігівської обласної державної адміністрації

Чернігівський обласний інститут післядипломної
педагогічної освіти імені К.Д. Ушинського

**ЗБІРНИК ЗАДАЧ ТА РОЗВ'ЯЗКІВ
II ЕТАПУ ВСЕУКРАЇНСЬКОЇ
УЧНІВСЬКОЇ ОЛІМПІАДИ
З ІНФОРМАТИКИ
2016-2017 навчального року**

Чернігів – 2017

Автори:

**Бондаренко С.М., Борзаков А.А., Дасюк А.М., Коваленко О.І.,
Фролов Д.А., Хрол Н.П.**

Рецензенти:

Горошко Ю. В., завідувач кафедри інформатики та обчислювальної техніки Чернігівського національного педагогічного університету імені Т. Г. Шевченка, доктор педагогічних наук, професор

Покришень Д. А., завідувач кафедри інформатики та інформаційно-комунікаційних технологій в освіті Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського, кандидат педагогічних наук, доцент

Загальна редакція:

Літош Ю. М., старший викладач кафедри інформатики та інформаційно-комунікаційних технологій в освіті Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського

Баранова О. Є., методист відділу природничо-математичних дисциплін Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського

Смірнова О.М., методист відділу природничо-математичних дисциплін Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського

Збірник задач та розв'язків II етапу Всеукраїнської учнівської олімпіади з інформатики 2016-2017 навчального року/ С. М. Бондаренко, А. А. Борзаков, А. М. Дасюк та ін.; за заг. ред. Ю. М. Літоша, О. Є. Баранової, О. М. Смірнової. – Чернігів: ЧОППО імені К. Д. Ушинського, 2017 – 31 с.

Збірник містить задачі та розв'язки II етапу Всеукраїнської учнівської олімпіади з інформатики, що проводилась у Чернігівській області у 2016-2017 навчальному році.

Матеріали розраховані на вчителів інформатики, слухачів курсів підвищення кваліфікації. Можуть бути використані учнями 7-11 класів для підготовки до олімпіад.

*Рекомендовано до друку
вченою радою Чернігівського обласного інституту
післядипломної педагогічної освіти імені К. Д. Ушинського
(протокол № 4 від 21.12.2017 р.)*

ЗМІСТ

Задачі I (першого) варіанту II етапу Всеукраїнської учнівської олімпіади з інформатики

| | |
|--|-----------|
| Задача А – Шоколадка | 4 |
| Задача В – Кросворд | 6 |
| Задача С – Цікава задача | 10 |
| Задача D – Увімкніть лампу | 13 |
| Задача Е – Свято в Ужляндії | 16 |

Задачі II (другого) варіанту II етапу Всеукраїнської учнівської олімпіади з інформатики

| | |
|--|-----------|
| Задача А – Басейн | 21 |
| Задача В – Гроші на сходах | 22 |
| Задача С – Пора року | 24 |
| Задача D – Пожежа | 25 |
| Задача Е – Шоколад | 27 |
| Список рекомендованої літератури..... | 30 |
| Список рекомендованих ресурсів мережі Інтернет..... | 31 |

Задачі I (першого) варіанту II етапу Всеукраїнської учнівської олімпіади з інформатики

Задача А-Шоколадка

| | |
|---------------------------|--------|
| Загальний бал: | 100 |
| Обмеження часу: | 100 мс |
| Обмеження реального часу: | 5 с |
| Обмеження пам'яті | 64М |

Степан вирішив пригостити однокласників шоколадками. Шоколадка коштувала N грн. З першого листопада вартість шоколадки збільшилась рівно на P відсотків. Визначте, скільки шоколадок зможе купити Степан на S грн після подорожчання.

Формат вхідних даних:

У першому рядку задано число N ($1 \leq N \leq 10^7$) - вартість шоколадки до подорожчання. У другому рядку - P ($0 \leq P \leq 100$) - величина подорожчання шоколадки у відсотках. В третьому рядку - S ($1 \leq S \leq 10^7$) - сума грошей, яка є у Степана.

Формат вихідних даних:

Виведіть одне число - кількість шоколадок, які може купити Степан.

Приклади

| Вхідні дані | Результат роботи |
|----------------|------------------|
| 25 5 100 | 3 |

Ідея розв'язання задачі

Хрол Н.П., учителя інформатики загальноосвітньої спеціалізованої школи I-III ст. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

1. Для точності обчислень переведемо вартість шоколадки n та суму грошей s , що є в Степана, в копійки.

2. Знайдемо вартість шоколадки t після подорожчання. Оскільки у нас вартість n в копійках, то дробову частину t відкидаємо.

3. Знайдемо кількість шоколадок k , які може купити Степан після подорожчання.

Розв'язання:

```
var n,s,p,k,t:real;
begin
  read(n);
  readln(p);
  readln(s);
  s:=s*100;
  n:=n*100;
  t:=n+trunc(n*p/100);
  k:=s/t;
  write(trunc(k));
end.
```

Ідея розв'язання задачі

Дасюка А.М., студента Київського національного університету ім. Т. Г. Шевченка, факультет кібернетики

В даній задачі необхідно знайти нову ціну шоколадки і визначити кількість шоколадок, які Степан може купити за свої кошти..

Розв'язання:

```
#include <iostream>
using namespace std;

int main()
{
  long double n, p;
  int s, answer;
  cin >> n >> p >> s;
  n += n / 100 * p; // Дізнаємось нову ціну шоколадки
  answer = s / n; // Знайдемо скільки можливо купити шоколадок за S
  гривень
  cout << answer;
  return 0;
}
```

Ідея розв'язання задачі

Борзакова А.А., учня 11 класу Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7 Прилуцької міської ради Чернігівської області

Розв'язання:

```
var n,m,s,k:extended;
```

```
begin
```

```
  read(n,p,s);
```

Знаходимо вартість шоколадки після подорожчання.

```
  n:=n+(n/100)*p;
```

Обчислимо кількість шоколадок, яку може купити Степан після подорожчання.

```
  k:=s /n;
```

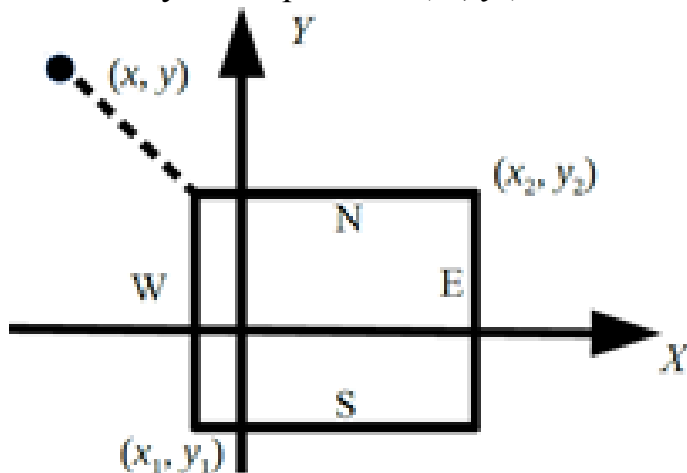
```
  write(trunc(k));
```

```
end.
```

Задача В-Напрямок руху

| | |
|---------------------------|--------|
| Загальний бал: | 100 |
| Обмеження часу: | 100 мс |
| Обмеження реального часу: | 5 с |
| Обмеження пам'яті: | 64М |

Степан влітку відпочиває у бабусі в селі. Особливо йому подобається купатись на сільському озері. Посередині озера плаває пліт, який має форму прямокутника. Сторони плоту спрямовані уздовж паралелей і меридіанів. Введемо систему координат, в якій вісь OX направлена на схід, а вісь OY - на північ. Нехай південно-західний кут плоту має координати (x_1, y_1) , північно-східний кут - координати (x_2, y_2) .



Степан знаходиться в точці з координатами (x, y) . Визначте, до якої сторони плоту (північної, південної, західної чи східної) або до будь-якого кута плоту (північно-західному, північно-східному, південно-західному, південно-східному) Степану потрібно плисти, щоб якомога швидше дістатися до плоту.

Формат вхідних даних:

Дано шість чисел в наступному порядку: x_1, y_1 (координати південно-західного кута плоту), x_2, y_2 (координати північно-східного кута плоту), x, y (координати Степана). Всі числа цілі і по модулю не перевершують 100. Гарантується, що $x_1 < x_2, y_1 < y_2, x \neq x_1, x \neq x_2, y \neq y_1, y \neq y_2$, координати Степана знаходяться поза плотом.

Формат вихідних даних:

Якщо Степану слід пливати до північної сторони плоту, програма повинна вивести символ «N», до південної - символ «S», до західної - символ «W», до східної - символ «E». Якщо Степану слід пливати до кута плоту, потрібно вивести один з наступних рядків: «NW», «NE», «SW», «SE».

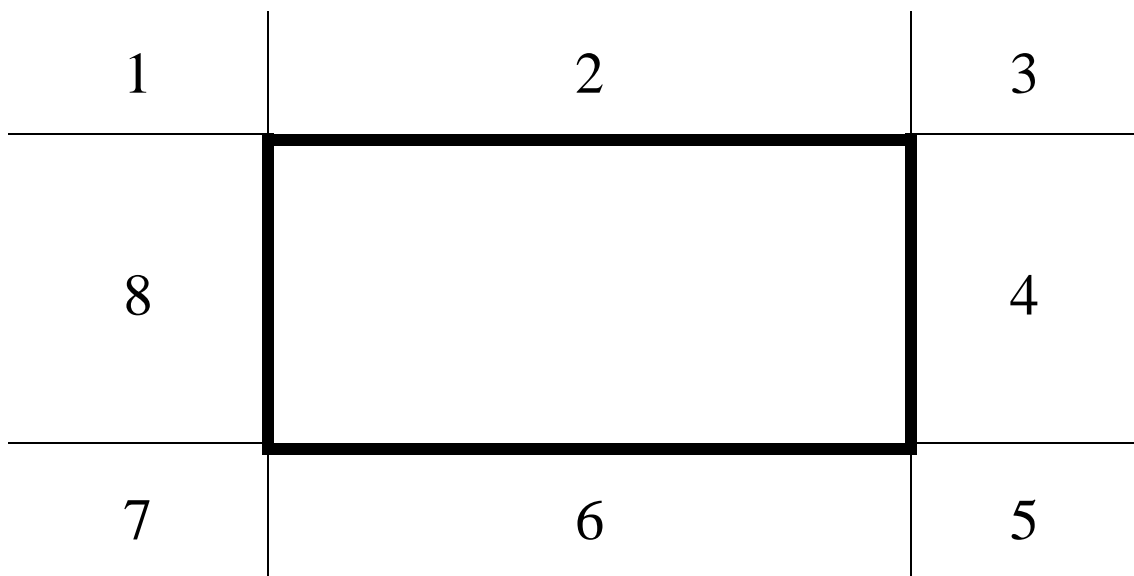
Приклади

| Вхідні дані | Результат роботи |
|-------------|------------------|
| -1 | NW |
| -2 | |
| 5 | |
| 3 | |
| -4 | |
| 6 | |

Ідея розв’язання задачі

Хрол Н.П., учителя інформатики загальноосвітньої спеціалізованої школи I-III ст. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

Пліт умовно розбиває площину на 8 частин. (див.мал.)



Очевидно, якщо точка з координатами (x, y) знаходиться у 2, 4, 6 або 8 частині площини, то найкоротшою відстанню до плоту буде перпендикуляр, проведений відповідно до північної, східної, південної або західної сторони.

Якщо точка з координатами (x, y) знаходиться у 1, 3, 5, 7 частинах площини, то найкоротшою відстанню буде відповідний кут плота.

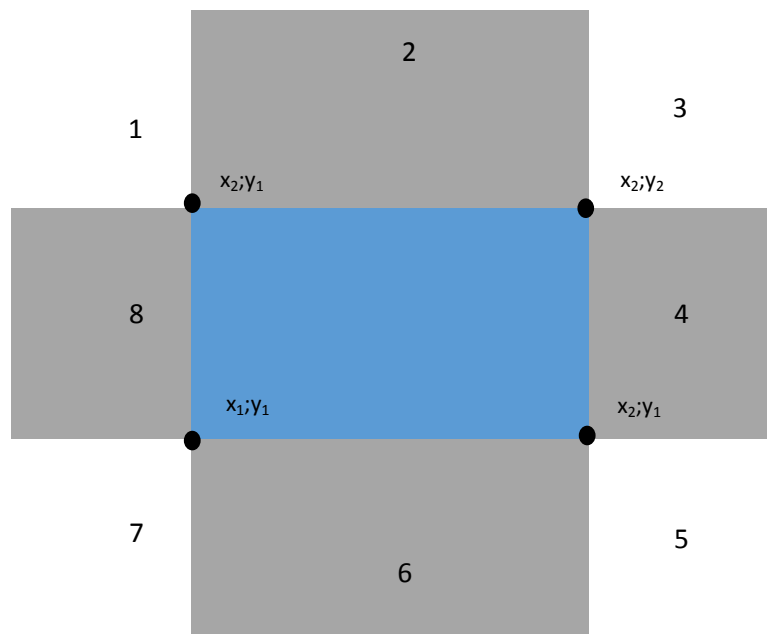
Розв’язання:

```
var x1,x2,x,y1,y2,y:integer;
begin
read(x1,y1,x2,y2,x,y);
if (x<x1) and (y>y1) and (y<y2) then write('W');
if (y>y2) and (x>x1) and (x<x2) then write('N');
if (x>x2) and (y>y1) and (y<y2) then write('E');
if (y<y1) and (x>x1) and (x<x2) then write('S');
if (x<x1) and (y>y2) then write('NW');
if (x>x2) and (y>y2) then write('NE');
if (x<x1) and (y<y1) then write('SW');
if (x>x2) and (y<y1) then write('SE');
end.
```

Ідея розв’язання задачі

Дасюка А.М., студента Київського національного університету ім. Т. Г. Шевченка, факультет кібернетики

Степан може знаходитися в восьми зонах відносно плоту. Якщо він знаходиться в 1, 3, 5 або 7 зоні, тоді мінімальною відстанню буде дистанція до одного з кутів плоту. В іншому випадку – необхідно пливати до сторони. Для визначення, в якій зоні знаходиться Степан, необхідно порівняти його координати з координатами плоту.



Розв’язання:

```
#include <iostream>
#include <cmath>
using namespace std;
```



```

double dist(int x1,int y1,int x2,int y2)
{
    return sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
}

int main()
{
    int x,y,x1,y1,x2,y2;
    cin >> x1 >> y1 >> x2 >> y2 >> x >> y;

    if(x >= x1 && x <= x2)
    {
        if(y > y2)
            cout<<"N";
        else
            cout<<"S";
    }else
    {
        if(y >= y1 && y <= y2)
        {
            if(x>x2)
                cout<<"E";
            else
                cout<<"W";
        }
        else
        {
            double nw = dist(x, y, x1, y2);
            double ne = dist(x, y, x2, y2);
            double sw = dist(x, y, x1, y1);
            double se = dist(x, y, x2, y1);
            double mn = min(min(sw, se), min(nw, ne));

            switch (mn){
                case nw: cout << "NW"; break;
                case ne: cout << "NW"; break;
                case sw: cout << "SW"; break;
                case se: cout << "SE"; break;
            }
        }
    }
    return 0;
}

```

Ідея розв'язання задачі

Борзакова А.А., учня 11 класу Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7 Прилуцької міської ради Чернігівської області

Простір навколо плоту можна розділити на 8 зон. Кожна зона описується своєю умовою належності, яка і визначає, в якій з них знаходиться Степан і, відповідно, напрямок руху.

Розв'язання:

```
var x,y,x1,y1,x2,y2:longint;  
begin  
  readln(x1,y1,x2,y2,x,y);  
  if (x<x1)and(y>y2)then writeln('NW');  
  if (x<x1)and(y<y1)then writeln('SW');  
  if (x>x2)and(y<y1)then writeln('SE');  
  if (x>x2)and(y>y2)then writeln('NE');  
  if (x>x1)and(x<x2)and(y>y2)then writeln('N');  
  if (x<x1)and(y<y2)and(y>y1)then writeln('W');  
  if (x>x2)and(y<y2)and(y>y1)then writeln('E');  
  if (x>x1)and(x<x2)and(y<y1)then writeln('S');  
end.
```

Задача С-Пакування речей

| | |
|---------------------------|--------|
| Загальний бал: | 100 |
| Обмеження часу: | 200 мс |
| Обмеження реального часу: | 5 с |
| Обмеження пам'яті: | 64М |

Степан збирає речі у відпустку. З собою в літак він може взяти ручну поклажу і багаж. Для ручної поклажі у нього є рюкзак, а для багажу - здоровенна валіза.

За правилами перевезення маса ручної поклажі не повинна перевищувати S кг, а багаж може бути будь-якої маси (за наднормативний багаж Степан готовий доплатити). Зрозуміло, найбільш цінні речі - ноутбук, фотоапарат, документи і т. д. - Степан хоче покласти в ручну поклажу.

Степан розклав усі свої речі в порядку зменшення їх цінності і починає складати найбільш цінні речі в рюкзак. Він діє в такий спосіб - бере найцінніший предмет, і якщо його маса не перевищує S , то кладе його в рюкзак, інакше кладе його до валізи. Потім він бере наступний за цінністю предмет, якщо його можна покласти в рюкзак, тобто якщо його маса разом з масою вже

покладених в рюкзак речей не перевищує S , то кладе його в рюкзак, інакше до валізи, і таким же чином процес триває для всіх предметів в порядку спадання їх цінності.

Визначте вагу рюкзачка і валізи після того, як Степан складе всі речі.

Формат вхідних даних:

Перший рядок вхідних даних містить число S ($1 \leq S \leq 2 \times 10^9$) – максимально дозволена вага рюкзачка. У другому рядку вхідних даних записано число N ($1 \leq N \leq 10^5$) - кількість предметів.

У наступних N рядках дано маси предметів, самі предмети перераховані в порядку спадання цінності (спочатку вказана маса найціннішого предмета, потім маса другого по цінності предмета і т. Д.). Всі числа натуральні, сума ваги всіх предметів не перевищує 2×10^9 .

Формат вихідних даних:

Виведіть два числа - вагу рюкзачка і вагу валізи (вага порожнього рюкзачка і валізи не враховується).

Приклади

| Вхідні дані | Результат роботи |
|-------------|------------------|
| 10 | 10 2 |
| 4 | |
| 6 | |
| 3 | |
| 2 | |
| 1 | |

Ідея розв'язання задачі

Хрол Н.П., учителя інформатики загальноосвітньої спеціалізованої школи I-III ст. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

Нехай $k1$ – вага рюкзачка і $k2$ – вага валізи.

Беремо перший предмет з вагою a і перевіряємо, якщо $k1+a$ не перевищує s , то збільшуємо $k1$ на a , інакше $k2$ збільшуємо на a .

Розв'язання:

```
var s,n,a,k1,k2,i:int64;  
begin  
  readln(s);  
  readln(n);  
  i:=1;
```

```

k1:=0;
k2:=0;
while i<=n do
  begin
  readln(a);
  if (k1+a<=s) then k1:=k1+a else k2:=k2+a;
  i:=i+1;
  end;
write(k1,' ',k2);
end.

```

Ідея розв'язання задачі

Дасюка А.М., студента Київського національного університету ім. Т. Г. Шевченка, факультет кібернетики

В даній задачі необхідно реалізувати принцип, який описаний в умові, а саме: беремо предмет, і якщо його маса не перевищує наявно-доступну масу, то кладемо його в рюкзак, інакше кладемо його до валізи.

Розв'язання:

```

#include <iostream>
using namespace std;
int main()
{
  long long s, n, h[100005], r=0, v=0;

  cin >> s >> n;
  for (int i = 0; i < n; ++ i)
    cin >> h[i];

  for (int i = 0; i < n; ++ i)
    if(h[i] <= s)
      s -= h[i], r += h[i];
    else
      v += h[i];

  cout<<r<<" "<<v;
  return 0;
}

```

Ідея розв'язання задачі

Борзакова А.А., учня 11 класу Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7 Прилуцької міської ради Чернігівської області

Весь алгоритм задачі вказано в умові: "бере ... предмет, і якщо його маса не перевищує S , то кладе його в рюкзак, інакше кладе його до валізи" . Потрібно лише додати цикл по всім предметам.

Розв'язання:

```
var v,b,n,s,i:longint;a:array [1..1000000] of longint;
begin
  read(s,n);
  for i:=1 to n do read(a[i]);
  b:=0;
  v:=0;
  for i:=1 to n do
    if b+a[i]<=s then b:=b+a[i] else v:=v+a[i];
  write(b,' ',v);
end.
```

Задача D-Розбиття на групи

| | |
|---------------------------|--------|
| Загальний бал: | 100 |
| Обмеження часу: | 100 мс |
| Обмеження реального часу: | 5 с |
| Обмеження пам'яті: | 64М |

Степан виписує на листочку усі цілі числа від 1 до N в кілька груп, при цьому якщо одне число ділиться на інше, то вони обов'язково будуть у різних групах.

Наприклад, якщо $N = 9$, то отримаємо 4 групи:

Перша група: 1.

Друга група: 2 3 7.

Третя група: 4 5 6.

Четверта група: 8 9.

Очевидно, що оскільки, будь-яке число ділиться на 1, то одна група завжди буде складатись тільки з числа 1, а от інші групи можуть бути створені різними способами.

Допоможіть Степану написати програму, яка визначає мінімальне число груп, на яке можна розбити усі числа від 1 до N у відповідності до наведеної вище умови.

Формат вхідних даних:

Перший рядок вхідних даних містить єдине число N ($1 \leq N \leq 10^9$).

Формат вихідних даних:

Виведіть одне число - знайдену мінімальну кількість груп.

Приклади

| Вхідні дані | Результат роботи |
|-------------|------------------|
| 9 | 4 |

Ідея розв'язання задачі

Хрол Н.П., учителя інформатики загальноосвітньої спеціалізованої школи I-III ст. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

Очевидно, що числа $2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$ і т.д. будуть лежати в різних групах. Всі інші числа можна розподілити між цими групами.

Отже, потрібно знайти найбільшу степінь k числа 2, таку що $2^k \leq n$.

Розв'язання:

```
var n,k,n2:int64;  
begin  
  read(n);  
  k:=0;  
  n2:=1;  
  while n2<=n do  
    begin  
      n2:=n2*2;  
      k:=k+1;  
    end;  
  write(k);  
end.
```

Ідея розв'язання задачі

Дасюка А.М., студента Київського національного університету ім. Т. Г. Шевченка, факультет кібернетики

Перша група буде містити в собі одне число – 1. Друга – прості числа. Третя – вдвічі збільшена друга, четверта – вдвічі збільшена третя, і т.д. Таким чином, ми можемо розбити всі числа на мінімальну кількість взаємно простих чисел. Отже, тоді ми будемо мати таку кількість груп: $\lceil \log_2 n \rceil$.

Розв'язання:

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;

    int t = 0, p = 1;
    while(p <= n) p *= 2, ++ t;

    cout<<t;

    return 0;
}
```

Ідея розв'язання задачі

Борзакова А.А., учня 11 класу Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7 Прилуцької міської ради Чернігівської області

Наприклад, якщо визначати кількість груп для послідовних чисел від 1 до 20, то можна побачити, що вона (кількість) є степенем числа 2. Таким чином для будь-якого числа N можна розглянути нерівність $2^k \leq N < 2^{k+1}$, і шуканою відповіддю буде число $k+1$.

Розв'язання:

```
var n,v,i,st:longint;
begin
    read(n);
    v:=1;
    st:=2;
    i:=2;
    while i<=n do
        begin
            i:=st*2;
            st:=i;
            inc(v);
        end;
    if n<>1 then write(v) else write(1);
end.
```

Задача Е-Новий податок в Ужляндії

| | |
|---------------------------|--------|
| Загальний бал: | 100 |
| Обмеження часу: | 500 мс |
| Обмеження реального часу: | 5 с |
| Обмеження пам'яті: | 64М |

Для поповнення бюджету в Ужляндії, відомій своїми гірськими туристичними маршрутами, ввели новий податок для туристів. Величина податку пропорційна довжині маршруту, але, оскільки маршрут проходить по горах і пройдено відстань, яка залежить від висоти спуску і підйому, підрахувати складно, податок визначається без урахування висоти, тобто величина податку пропорційна горизонтальному переміщенню, що здійснила туристична група. Крім того, в силу старовинного звичаю усі туристичні групи повинні переміщатися по горах Ужляндії строго із заходу на схід.

Турфірма хоче заощадити на податку, тому вона хоче розробити туристичний маршрут з мінімальною величиною податку. При цьому, оскільки маршрут є гірським, він повинен містити підйом в гору і спуск з гори, тобто на маршруті має бути точка, яка знаходиться строго вище початку і кінця маршруту.

Турфірма склала карту гір Ужляндії, що містить інформацію про висоту гір при пересуванні із заходу на схід. Висоти гір виміряні в точках через однакові відстані. Знайдіть на цій карті гір туристичний маршрут мінімальної довжини, що задовольняє умові наявності підйому і спуску.

Формат вхідних даних:

Перший рядок вхідних даних містить число N - кількість точок на карті гір Ужляндії. Наступні N рядків містять інформацію про висоту гір в даних N точках при русі із заходу на схід. Всі числа натуральні, не перевищують 10^5 .

Формат вихідних даних:

Виведіть два числа - номер точки початку маршруту і номер точки закінчення маршруту. Точки нумеруються від 1 до N . Якщо маршруту, що задовольняє умовам, не існує, програма повинна вивести одне число 0. Якщо знайдеться декілька маршрутів, то вивести той, що починається лівіше.

Пояснення до прикладів:

Перший приклад: Дано 7 точок з висотами 18, 10, 15, 20, 20, 10, 3. Найкоротший маршрут, який містить підйом і спуск, - це 15, 20, 20, 10. Він починається в точці номер 3 і закінчується в точці номер 6.

Другий приклад: Висота гір монотонно спадає, тому шуканого маршруту не існує.

Приклади

| Вхідні дані | Результат роботи |
|--|------------------|
| 7 18 10 15 20 20 10 3 | 3 6 |
| 3 9 8 5 | 0 |

Ідея розв'язання задачі

Хрол Н.П., учителя інформатики загальноосвітньої спеціалізованої школи I-III ст. фізико-математичного профілю № 12 м. Чернігова, старшого вчителя

Нехай масив A складається з елементів, що містять інформацію про висоту гір в даних N точках при русі із заходу на схід.

Потрібно розглянути випадки:

1. Якщо існує послідовність з трьох елементів масиву A таких, що $A[i-1] < A[i]$ і $A[i] > A[i+1]$, то це буде найкоротший маршрут.
2. Якщо існують послідовності з елементів масиву A такі, що $A[i-1] < A[i]$, $A[i] = A[i+1]$, $A[i] = A[i+2]$, ... $A[i] < A[i+k]$, де $k \leq n$, то серед них потрібно знайти найкоротшу.

Розв'язання:

```
var A:array [1..100000] of int64;  
    i,n,s,k,j,f:longint;  
begin  
  readln(n);  
  for i:=1 to n do  
    readln(A[i]);  
  s:=1000000;  
  for i:=2 to n-1 do  
    begin  
      if (A[i]>A[i-1]) and (A[i]>A[i+1]) then begin  
        write(i-1,' ',i+1);  
        exit;
```

```

end;
if (A[i]>A[i-1]) and (A[i]=A[i+1]) then begin
    k:=i-1;
    j:=i;
    while (A[j]=A[i]) and (j<n) do
        begin
            j:=j+1;
        end;
    if (j=n) and (A[j]=A[i]) then begin
        if s<1000000 then write(f,' ',f+s-1) else
write(0);
        exit;
    end;
    if A[j]<A[i] then begin
        if s>(j-k+1) then
            begin
                s:=j-k+1;
                f:=k;
            end;
        end;
    end;
end;
if s<1000000 then write(f,' ',f+s-1) else write(0);
end.

```

Ідея розв'язання задачі

Дасюка А.М., студента Київського національного університету ім. Т. Г. Шевченка, факультет кібернетики

Головна ідея розв'язку цієї задачі – змінити спосіб збереження карти. Карту можна зберігати за таким принципом: висота вершини, початковий індекс, кінцевий індекс. Для такого перетворення необхідно зробити один прохід в циклі, тобто воно займає лінійний час. Далі просто перебираємо нашу карту на наявність вершини, і даємо відповідь.

Розв'язання:

```

#include <iostream>
#include <vector>
using namespace std;

int main()
{
    int n;
    cin>>n;

```

```

vector <int> a(n);
vector <pair<int,pair<int,int> > > without_repeats;
for(int i=0;i<n;i++)
    cin>>a[i];

for(int i=0;i<n;i++)
{
    int j=i+1;
    while(a[i]==a[j])j++;
    without_repeats.push_back(make_pair(a[i],
                                        make_pair(i,j-1)));
    i=j - 1;
}

int mn=1e9,c,b;
bool f=false;

for (int i = 1; i < without_repeats.size() - 1; ++ i)
    if(without_repeats[i - 1].first < without_repeats[i].first &&
without_repeats[i].first > without_repeats[i + 1].first)
        {
            f = true;
            if (without_repeats[i + 1].second.first - without_repeats[i -
1].second.second < mn)
                mn = without_repeats[i + 1].second.first - without_repeats[i -
1].second.second,
                c = without_repeats[i - 1].second.second,
                b = without_repeats[i + 1].second.first;
        }

if(!f)return cout << 0, 0;
cout << c + 1 << " " << b + 1;
return 0;
}

```

Ідея розв'язання задачі

Борзакова А.А., учня 11 класу Прилуцької загальноосвітньої школи I-III ступенів № 7 Прилуцької міської ради Чернігівської області

В зворотньому порядку шукаємо перший можливий початок шляху. Потім дивимося, чи він є менший за знайдений попередній шлях. Починаємо з кінця масиву через те, що нам потрібно знайти перший мінімальний.

Розв'язання:

```
var i,p,k,mp,mk,n:longint;a:array [0..100001] of longint;
begin
  read(n);
  for i:=1 to n do
    read(a[i]);
  p:=0;
  k:=0;
  mk:=100000;
  mp:=0;
  Цикл з кінця масиву до початку
  for i:=n downto 1 do
    begin
      Перевірка на кінець шляху
      if (k=0) then if (a[i]<a[i-1]) then k:=i else continue;
      Перевірка на початок шляху
      if (a[i]<a[i+1])and(k-i>=1) then p:=i;
      Перевірка на мінімальну відстань
      if (k-p)<=(mk-mp) then begin mk:=k;mp:=p;end;
      Перевірка останніх ітерацій на шлях
      if (a[i]<a[i-1])and(i>2) then begin k:=i;p:=0;end;
      end;
      Перевірка на наявність шляху після всіх ітерацій
      if mp<>0 then write(mp,' ',mk) else write(0);

    end.
```

Задачі II (другого) варіанту II етапу Всеукраїнської учнівської олімпіади з інформатики

Задача А – Басейн

Ромчик влітку їздив на море. На березі хлопчик помітив порожній басейн, який вміщає V літрів води. У Ромчика з собою є відро об'ємом 4 літри. Допоможіть йому обчислити скільки разів потрібно сходити за водою, щоб наповнити басейн.

Вхідні дані:

Ціле число V – об'єм басейну в літрах. $80 \leq V \leq 10000$.

Вихідні дані:

Ціле число k – кількість ходок за водою, щоб наповнити басейн.

Приклад:

| Вхідні дані | Вихідні дані |
|-------------|--------------|
| 80 | 20 |
| 205 | 52 |

Ідея розв'язання задачі

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста

Поділимо націло об'єм басейну на об'єм відра. Якщо матимемо залишок, то до знайденого числа треба додати 1.

Розв'язання:

```
var v,k:integer;
begin
  read(v);
  k:=v div 4;
  if v mod 4 <>0 then inc(k);
  write(k);
end.
```

Ідея розв'язання задачі

Фролова Д.А., учня 10 класу спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова;

Коваленко О.І., учителя-методиста, учителя інформатики спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова

Поділимо v на 4, таким чином ми дізнаємось скільки повних відер потрібно перенести до басейну, але якщо остача від ділення v на 4 не дорівнює 0, то Ромчик повинен перенести ще одне напівпuste відро.

Розв'язання:

```
#include <iostream>

using namespace std;

int main()
{
    int v;
    cin >> v;
    if(v % 4)
        cout << v / 4 + 1;
    else
        cout << v / 4;

    return 0;
}
```

Задача В – Гроші на сходах

Ромчик піднімався сходами. На першій же сходинці він знайшов 1 копійку, на другій лежало 2 копійки. Якби на кожній наступній сходинці лежало на одну копійку більше ніж на попередній, то скільки Ромчику потрібно пройти сходинок n , щоб назбирати k гривень?

Вхідні дані:

Ціле число k – кількість гривень, які Ромчику потрібно зібрати. $1 \leq k \leq 50000$.

Вихідні дані:

Ціле число n – кількість сходинок.

Приклад:

| Вхідні дані | Вихідні дані |
|-------------|--------------|
| 1797 | 599 |

Ідея розв'язання задачі

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста

Найпростіший шлях – організувати цикл згідно умови задачі. Також можна застосувати формулу суми арифметичної прогресії, згідно якої і розміщені гроші на сходах.

Розв’язання:

```
var i,n,s:longint;
begin
  read(n);
  i:=1;s:=1;
  while s<n do
  begin
    inc(i);
    s:=s+i;
  end;
  if s>=n then write(i) else write (i+1);
end.
```

Ідея розв’язання задачі

Фролова Д.А., учня 10 класу спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова;

Коваленко О.І., учителя-методиста, учителя інформатики спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова

В даній задачі необхідно запам’ятовувати скільки грошей в Ромчика на даній сходинці і виконувати це, допоки він не назбирав необхідну суму.

Розв’язання:

```
#include <iostream>

using namespace std;

int main()
{
  int n, b;
  cin >> n;
  for(int i = 1; i <= n; ++ i);
  {
    b += i;
    if(b>=n)
      return cout << i, 0;
  }
  return 0;
}
```

Задача С – Пору року

Визначити назву пору року за заданим номером місяця, використовуючи складені умови.

Вхідні дані:

Одне число - номер місяця.

Вихідні дані:

Для весняних місяців вивести Spring, для літніх - Summer, для осінніх – Autumn і для зимових - Winter.

Приклад:

| Вхідні дані | Вихідні дані |
|-------------|--------------|
| 5 | Spring |

Ідея розв'язання задачі

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7, учителя-методиста

Розв'язання:

```
var n:integer;
begin
  readln(n);
  case n of
    12,1,2:write('Winter');
    3,4,5:write('Spring');
    6,7,8:write('Summer');
    9,10,11:write('Autumn');
  end;
end.
```

Ідея розв'язання задачі

Фролова Д.А., учня 10 класу спеціалізованої загальноосвітньої школи № 2 І-ІІІ ст. з поглибленим вивченням іноземних мов м. Чернігова;

Коваленко О.І., учителя-методиста, учителя інформатики спеціалізованої загальноосвітньої школи № 2 І-ІІІ ст. з поглибленим вивченням іноземних мов м. Чернігова

Для цієї задачі порібно знати, який місяць належить до відповідної пору року.

Розв'язання:

```
#include <iostream>
using namespace std;
```



```

int main()
{
    int a;
    cin >> a;
    if(a == 1 || a == 2 || a == 12)
        cout<<"Winter";
    if(a == 3 || a == 4 || a == 5)
        cout<<"Spring";
    if(a == 6 || a == 7 || a == 8)
        cout<<"Summer";
    if(a == 9 || a == 10 || a == 11)
        cout<<"Autumn";
    return 0;
}

```

Задача D – Пожежа

Трюм баржі, що має форму квадрата зі стороною C ($C = 1..1000$), розбитий на занумеровані зліва направо по рядкам одиничні квадрати (комірки), в яких міститься вогненебезпечний вантаж. Рівно опівночі внаслідок грози спалахнула комірка з номером N . Кожна комірка палає 1 хв, а потім пожежа перекидається у всі сусідні комірки через спільні сторони. Скільки годин (G) і хвилин (H) тривала пожежа, якщо увесь вантаж згорів?

Вхідні дані:

Два натуральні числа C і N ($N = 1..C^2$).

Вихідні дані:

Два невід’ємні цілі числа G і H (кількості годин і хвилин).

Приклад:

| Вхідні дані | Вихідні дані |
|-------------|--------------|
| 3 5 | 0 3 |

Ідея розв’язання задачі

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи I-III ступенів № 7, учителя-методиста

Шукана кількість часу дорівнює сумі найдовших відстаней від даної клітинки до стінок трюму по горизонталі та вертикалі. Також потрібно не забути перевести знайдений час у години та хвилини.

Розв’язання:

```

var c,n,x,y,maxx,maxy,h,g:longint;
begin
  read(c,n);
  x:= (n div c);
  y:= (n mod c);
  if y=0 then begin dec(x);y:=c;end;
  if x>c-x-1 then maxx:=x else maxx:=c-x-1;
  if y-1>c-y then maxy:=y-1 else maxy:=c-y;
  g:=(maxx+maxy+1) div 60;
  h:=(maxx+maxy+1) mod 60;
  write(g,' ',h)
end.

```

Ідея розв'язання задачі

Фролова Д.А., учня 10 класу спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова;

Коваленко О.І., учителя-методиста, учителя інформатики спеціалізованої загальноосвітньої школи № 2 I-III ст. з поглибленим вивченням іноземних мов м. Чернігова.

Останньою клітинкою, яка спалахне, буде одна з крайніх. Визначимо координати початкової клітинки і найвіддаленішу до неї крайню клітинку. Сума модулів різниць між їх координатами і буде відповіддю на задачу.

Розв'язання:

```

#include <iostream>

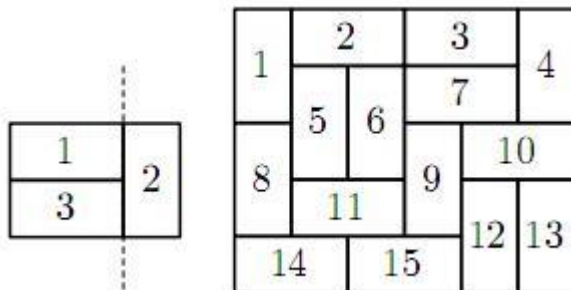
using namespace std;

int main()
{
  int c, n;
  cin >> c >> n;
  int x = n / c + 1, y = n % c;
  if(c == 1) return cout << "0 1", 0;
  int h = 1 + max(c - x, x - 1) + max(c - y, y - 1);
  cout << h / 60 << " " << h % 60;
  return 0;
}

```

Задача Е – Шоколад

Петрик і Маша дуже люблять шоколад. Нещодавно Петрик купив шоколадку і тепер хоче поділитися нею з Машею. Шоколадка представляє собою прямокутник $n \times m$, яка повністю складається з маленьких шоколадних дольок — прямокутників 2×1 .



Петрик ділить шоколадку на дві частини, розламуючи її вдовж деякої прямої, паралельної одному з країв шоколадки. Ні Петрик, ні Маша не люблять поламані дольки, тому Петрик хоче розламати шоколад так, щоб ні одна долька не була пошкоджена.

Допоможіть Петрику поділитися шоколадкою з Машею.

Вхідні дані:

У першому рядку вхідного файлу два цілих числа n і m ($1 \leq n, m \leq 20$; хоча б одне з чисел n і m — парне). Далі слідує n рядків по m чисел у кожному — номери дольок, у які входять відповідні кусочки шоколадки. Дольки мають номери від 1 до $(n \cdot m)/2$, і ніякі дві дольки не мають однакових номерів.

Вихідні дані:

У вихідний файл виведіть "Yes", якщо Петрик може розламати шоколадку, не пошкодивши дольки. Інакше виведіть "No".

Приклад:

| Вхідні дані | Вихідні дані |
|--|-----------------|
| Sample 1 2 3 1 1 2 3 3 2 | Sample 1 Yes |
| Sample 2 5 6 1 2 2 3 3 4 1 5 6 7 7 4 8 5 6 9 10 10 8 11 11 9 12 13 14 14 15 15 12 13 | Sample 2 No |

Ідея розв'язання задачі

Бондаренка С. М., учителя математики та інформатики Прилуцької загальноосвітньої школи І-ІІІ ступенів № 7, учителя-методиста

Плитку можна розділити на дві частини, щоб ні одна долька не була пошкоджена, якщо вздовж якоїсь вертикальної чи горизонтальної лінії по обидва боки розміщені різні за номером плитки шоколаду.

Розв'язання:

(94 бали із 100)

```
var n,m,i,j:integer; a:array[1..20,1..20]of integer; flag:boolean;
begin
  read(n,m);
  for i:=1 to n do
    for j:=1 to m do read(a[i,j]);
  for j:=1 to m-1 do
    begin
      flag:=true;
      for i:=1 to n do
        if a[i,j]=a[i,j+1] then begin flag:=false;break;end;
      if flag then begin writeln('Yes');exit;end;
    end;
  for i:=1 to n-1 do
    begin
      flag:=true;
      for j:=1 to m do
        if a[i,j]<>a[i+1,j] then begin flag:=false;break;end;
      if flag then begin writeln('Yes');exit;end;
    end;
  write('No');
end.
```

| | | |
|---|-----|---|
| 1 | 3 3 | |
| 1 | 4 | 6 |
| 2 | 4 | 6 |
| 2 | 5 5 | |

Ідея розв'язання задачі

Фролова Д.А., учня 10 класу спеціалізованої загальноосвітньої школи № 2 І-ІІІ ст. з поглибленим вивченням іноземних мов м. Чернігова;

Коваленко О.І., учителя-методиста, учителя інформатики спеціалізованої загальноосвітньої школи № 2 І-ІІІ ст. з поглибленим вивченням іноземних мов м. Чернігова

Пробіжимося по шоколадці і будемо пробувати її ділити вздовж та поперек кожного рядочка та стовпчика.

Розв'язання:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int n, m;
    cin >> n >> m;
    int a[30][30];
    for (int i = 0; i < n; ++ i)
        for (int j = 0; j < m; ++ j)
            cin >> a[i][j];
    for (int i = 0; i < n - 1; ++ i)
    {
        bool f = true;
        for (int j = 0; j < m; ++ j)
            if(a[i][j]==a[i + 1][j])
                f = false;
        if(f)return cout << "Yes", 0;
    }
    for (int j = 0; j < m - 1; ++ j)
    {
        bool f = true;
        for (int i = 0; i < n; ++ i)
            if(a[i][j + 1]==a[i][j])
                f = false;
        if(f)return cout << "Yes", 0;
    }
    cout << "No";
    return 0;
}
```

Список рекомендованої літератури

1. Збірник задач та розв'язків III етапу Всеукраїнських учнівських олімпіад з інформатики 2012-2013 навчального року / укл. О.Є. Баранова, Ю.М. Літош, В.В. Зуб, С.М. Бондаренко, О.М. Смірнова. – Чернігів: ЧОІППО імені К.Д. Ушинського, 2015. – 50 с.
2. Збірник задач та розв'язків III етапу Всеукраїнських учнівських олімпіад з інформатики 2013-2014 навчального року / укл. О.Є. Баранова, Ю.М. Літош, В.В. Зуб, С.М. Бондаренко, О.М. Смірнова. – Чернігів: ЧОІППО імені К.Д. Ушинського, 2015. – 60 с.
3. Збірник задач та розв'язань III етапу Всеукраїнської учнівської олімпіади з інформатики 2014-2015 навчального року/ С. М. Бондаренко, Н. І. Бондарчук, С. О. Гах та ін.; за заг. ред. О. Є. Баранової, Ю. М. Літоша. – Чернігів: ЧОІППО імені К. Д. Ушинського, 2016. – 71 с.
4. Збірник задач та розв'язків II етапу Всеукраїнської учнівської олімпіади з інформатики 2014-2015 навчального року/ укл. С.М. Бондаренко, В.В. Зуб, Ю.М. Літош. – Чернігів: ЧОІППО імені К.Д. Ушинського, 2015. – 35 с.
5. Збірник задач та розв'язків II етапу Всеукраїнських учнівських олімпіад з інформатики 2015-2016 навчального року / укл. В. М. Харченко. – Ніжин: НДУ імені Миколи Гоголя, 2016. – 54 с.
6. Збірник задач та розв'язань із програмування / С.М. Бондаренко, В.В. Зуб, О.І. Коваленко та ін. ; за заг. ред. О. Є. Баранової, Ю. М. Літоша. – Чернігів: ЧОІППО імені К. Д. Ушинського, 2016. – Ч.1. – 40 с.
7. Збірник задач та розв'язків із програмування / Н. П. Хрол, В. М. Харченко, Л. Б. Коваленко та ін.; за заг. ред. Ю. М. Літоша, О. Є. Баранової, О. М. Смірнової. – Чернігів: ЧОІППО імені К. Д. Ушинського, 2016. – Ч.2. – 71 с.
8. Харченко В.М. Готуємо до олімпіади з інформатики / В. М. Харченко. – Ніжин: НДУ ім. М. Гоголя, 2015. – Ч. 2. – 121 с.

Список рекомендованих ресурсів мережі Інтернет

1. Чернігівська обласна Інтернет-школа «Юний програміст» [Електронний ресурс]. – Режим доступу: <http://choippo.cn.sch.in.ua/> - Назва з екрану.
2. ACM-Контестер: Український портал ACM-спільноти [Електронний ресурс]. – Режим доступу: <http://acm.lviv.ua/> – Назва з екрану.
3. Дистанційна підготовка по інформатиці [Електронний ресурс]. – Режим доступу: <http://informatics.mcsme.ru/moodle/> – Назва з екрану.
4. Дніпропетровські олімпіади з інформатики [Електронний ресурс]. – Режим доступу: <http://oi.dp.ua/> – Назва з екрану.
5. Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України [Електронний ресурс]. – Режим доступу: <http://e-olymp.com/> – Назва з екрану.
6. Київські учнівські олімпіади з інформатики. Проведення ві результати [Електронний ресурс]. – Режим доступу: <http://kievoi.ipro.kubg.edu.ua/> – Назва з екрану.
7. Матеріали Всеукраїнських учнівських олімпіад з інформатики [Електронний ресурс]. – Режим доступу: <http://matholymp.org.ua/contests/types/olympiads/informatics/> – Назва з екрану.
8. Центр підтримки та проведення олімпіад школярів з використанням можливостей Internet [Електронний ресурс]. – Режим доступу: <http://www.olymp.vinnica.ua/> – Назва з екрану.
9. Школа програміста [Електронний ресурс]. – Режим доступу: <http://acmp.ru/> – Назва з екрану.
10. Timus Online Judge: архив задач с проверяющей системой [Електронний ресурс]. – Режим доступу: <http://acm.timus.ru/> – Назва з екрану.