

Збільшити на 2

Задано одновимірний масив A цілих чисел. Збільшити на 2 кожний невід'ємний елемент масиву.

Вхідні дані

У першому рядку задано натуральне число h - кількість елементів масиву ($h \leq 100$). У другому рядку через проміжок задано самі елементи масиву, значення кожного з яких за модулем не перевищує 100.

Вихідні дані

В єдиному рядку вивести через проміжок h чисел: нові значення елементів масиву, у тому ж порядку, в якому їх було задано.

Вхідні дані

4
1 2 3 4

Вихідні дані

3 4 5 6

Пояснення: При виведенні результатів роботи числа повинні бути відокремлені одним пропуском, після останнього елементу пропуск повинен бути відсутнім.

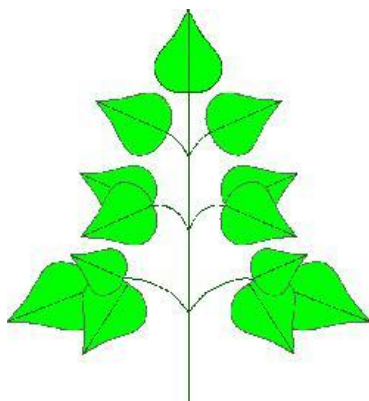
Розв'язок

Вводимо число. Використовуючи цикл **for** вводимо елементи масиву, порівнявши їх з нулем виводимо додатні числа збільшені на 2, від'ємні без зміни.

```
var a,i,h:integer;  
begin  
  readln(h);  
  for i:=1 to h do  
    begin  
      read(a);  
      if a>=0 then write(' ',a+2)  
        else write (' ',a);  
    end;  
  end;  
end.
```

Юний садівник

Мама попросила Васю полити всі молоді деревця у саду. Вася знає, що поки дерева маленькі, їх потрібно дуже добре поливати. А ось скільки поливати - невідомо. Але Вася - дуже розумний хлопчик. Він уважно прочитав весь підручник ботаніки для середньої школи і вияснив, що полив прямо пропорційний кількості листочків на дереві. Для гарного росту дерев достатньо виливати під дерево щоденно по одному літру води для кожного листка.



На щастя Васі виявилось, що листки на деревах ростуть ярусами, причому на верхньому ярусі два листка, на другому - чотири, на наступному - шість, і так далі, на кожному наступному ярусі на два листки більше у порівнянні з попереднім. А на самій верхушці росте ще один листочок. Хитрий Вася послав молодшу сестричку Машеньку підрахувати кількість ярусів на кожному дереві, а Вас просить написати програму, яка для кожного дерева обрахує кількість літрів води для його поливу.

Вхідні дані

Кількість ярусів n ($0 \leq n \leq 1000$) на дереві.

Вихідні дані

Вивести кількість літрів води для поливу цього дерева.

Вхідні дані

3

Вихідні дані

13

Розв'язок

Для розв'язку цієї задачі потрібно знайти суму арифметичної прогресії, де $n=n$, $a_1=2$ і $d=2$ добавивши до неї 1 (листок з верхівки). Для цього можна скористатися формулою суми арифметичної прогресії $S_n = \frac{2a_1+d(n-1)}{2}n \rightarrow n^2 + n$, або за допомогою циклу n раз додати номери ярусів збільшені у два рази

```
var i,n,s:int64;  
begin  
  readln(n); s:=1;  
  for i=1 to n do  
    s:=s+i*2;  
  writeln(s)  
end.
```

Юні програмісти



Відомо, що у школі не менше ніж k_1 учнів, але не більше ніж k_2 учнів. Також відомо, що кожен хлопчик дружить з n дівчатками, а кожна дівчинка з m хлопчиками. Яка мінімальна кількість учнів може бути у школі, і скільки у школі хлопчиків та дівчаток?

Юні програмісти, як Ви бачите, до цього часу розв'язують цю задачку. Допоможіть їм.

Вхідні дані

У першому рядку вхідного файлу знаходиться 4 числа, відокремлених пропусками: k_1 , k_2 , n і m . Всі вхідні дані натуральні числа, які не перевищують 10000, $k_1 \leq k_2$.

Вихідні дані

У єдиному рядку вивести через пропуск три числа: спочатку кількість учнів у школі, потім кількість хлопчиків та дівчаток. Гарантується, що вхідні дані коректні і відповідь завжди існує.

Вхідні дані

20 30 4 5

Вихідні дані

27 15 12

Розв'язок

Збільшуємо кількість дівчаток (g) на 1, поки кількість хлопчиків (b) по формулі $b := g * m / n$ не стане цілим числом і сума хлопчиків та дівчаток буде більше або рівна мінімальній кількості учнів.

```
var k1, k2, n, g, m: int64;  
    b: real;  
begin  
  readln(k1, k2, n, m);  
  for g:=n to k2 do  
    begin  
      b:= g*m/n;  
      if (Frac(b)=0) and ((b+g)>=k1) then break;  
    end;  
  writeln(b+g, ' ', b, ' ', g);  
end.
```

Бакси в банці

Папа Карло подарував Буратіно 1 долар на його перший день народження, а заощадливий Буратіно поклав подарунок у банку. Кожного наступного року папа Карло подвоював свій попередній подарунок і додавав до нього стільки доларів, скільки років виповнилось Буратіно, а той в свою чергу продовжував складати бакси у банку. На який N-й день народження у банці буде не менш ніж S доларів?

Вхідні дані

Єдине число - значення S. $1 \leq S \leq 2^{40}$.

Вихідні дані

Шукане значення N.

Вхідні дані

15

Вихідні дані

3

Розв'язок

В даній задачі виконуємо повторення порівнюючи задану суму(S) із шуканою(S1) і збільшуючи її кожен раз на величину подарунку $p=p*2+n$, рахуємо рік(n), коли це станеться.

```
var s,s1,n,p:int64;
```

```
begin
```

```
  readln(s);
```

```
  s1:=1;
```

```
  p:=1;
```

```
  n:=1;
```

```
while s1<s do
```

```
  begin
```

```
    inc(n);
```

```
    p:=p*2+n;
```

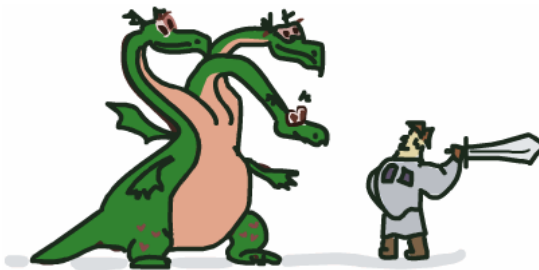
```
    s1:=s1+p;
```

```
  end;
```

```
writeln(n);
```

```
end.
```

Змій Горинич



В деякому царстві жив Змій Горинич. У нього було N голів та M хвостів. Іван-царевич вирішив знищити губителя людських душ, для чого йому його кума Баба Яга подарувала чарівний меч, оскільки тільки ним можна вбити Змія Горинича. Якщо відрубати одну голову, то на її місці виростає нова, якщо відрубати хвіст, то замість нього виростає 2 хвосту. Якщо відрубати два хвосту, то виростає 1 голова, і тільки коли зрубати 2 голови, то не

виростає нічого. Змій Горинич гине тільки в тому випадку, коли йому відрубати всі голови і всі хвосту. Визначити мінімальну кількість ударів мечем, потрібну для знищення Змія Горинича.

Вхідні дані

Два числа N, M ($0 \leq N, M \leq 1000$).

Вихідні дані

Єдине число – мінімальна кількість ударів мечем, або -1, якщо знищити Змія Горинича неможливо.

Вхідні дані

3 3

Вихідні дані

9

Розв'язок

При рішенні даної задачі розглянуто декілька випадків:

- Змія Горинича неможливо вбити – це коли в нього непарна кількість голів і нема жодного хвоста, якщо це так то виводим «-1», інакше переходимо до другого пункту рішення;

- Змія Горинича можна вбити – це коли у Змія є хоча б один хвіст або парна кількість голів. В такому випадку ми працюємо всередині циклу, при кожному проході якого перевіряється чи виконуються описані вище умови:

while ((N>0)or(M>0)) **do** {цикл буде виконуватися до тих пір, поки є хоча б одна голова чи хвіст } Горинича можна вбити тільки тоді коли при відрубуванні всіх хвостів залишиться парна кількість голів. Звідси, необхідно, щоб у Змія була парна кількість хвостів і сума голів і хвостів поділена на два, була парним числом. Тому будемо працювати по даному алгоритму:

1. Якщо кількість хвостів непарна, то відрубаємо один хвіст, на місці якого виросте два нових, і збільшуємо лічильник ударів на 1;
2. Якщо кількість голів непарна і хвостів більше одного, то відрубуємо два хвости, тим самим збільшуємо кількість голів на 1 і збільшуємо лічильник ударів;
3. Якщо кількість голів і хвостів парна і якщо кількість хвостів при діленні на два дає парне число, то ми збільшуємо лічильник на $(M/2 + ((N+(M/2))/2))$, і прирівнюємо кількість хвостів і голів до нуля;
4. Якщо якась з умов пункту 3 не виконується, то збільшуємо кількість хвостів на 1, і збільшуємо лічильник на 1;
5. Повторюємо алгоритм до тих пір поки не вб'ємо Горинича.

var N, M, count:integer;

begin

count:=0;

readln(N,M);

if ((N mod 2 <>0) **and** (M=0)) {перевірка чи можна вбити Горинича }

then **writeln**(' -1'){вивести " -1" }

else

begin

while ((N>0)or(M>0)) **do** {цикл буде виконуватися до тих пір, поки є хоча б одна голова чи хвіст }

begin

if (M mod 2=1) {якщо число хвостів непарне, то відрубавши один хвіст збільшуємо їх кількість на 1 }

then

begin

inc(count); {збільшення лічильника відрізів на 1 }

inc(M);

end;

if ((N mod 2=1) **and** (M>=2)) {якщо голів непарна кількість, то відрубуємо два хвости(M) і збільшуємо кількість голів(N) на 1 }

then

begin

M:=M-2;

inc(N);

inc(count);

end;

if (M mod 2=0) **and** (N mod 2=0) **and** (((N+(M div 2)) mod 2)=0) {якщо число голів і хвостів непарне, і якщо кількість хвостів при діленні на два дає парне число }

then begin

count:=count+(M div 2)+((N+(M div 2)) div 2); {то збільшуємо лічильник надрізів }

N:=0;

M:=0; {і обнуляємо кількість голів і хвостів для виходу з циклу }

end

else {якщо не виконується хоча б одна з умов минулого розгалуження, то кількість хвостів збільшуємо на 1 }

begin

inc(M);

```

        inc(count);
    end;
end;
writeln(count);{виводим кількість ударів}
end;
end.

```

Постійна сума цифр

Знайти кількість двохзначних чисел, які не змінюють свою суму цифр при множенні числа на однозначне ціле число N ($N=0..9$).

Вхідні дані: Ціле число N . $0 \leq N \leq 9$.

Вихідні дані: Відповідь до задачі.

Вхідні дані

2

Вихідні дані

10

Пояснення: При множенні двоцифрових чисел на 2 не змінюють суму цифр такі числа: 18 27 36 45 54 63 72 81 90 99.

Розв'язок

Потрібно перебрати всі двозначні числа та знайти добутки цих чисел на задане число. Знайти суму цифр(a) двозначного числа та суму цифр(b) отриманого числа(цикл while). Якщо суми однакові то збільшуємо лічильник.

```
var i, a, b, k, n: integer;
```

```
begin
```

```
  readln (n); {вводимо число на яке будуть множитися всі двозначні числа}
```

```
  k:=0;
```

```
  for i:=10 to 99 do{цикл от 10 до 99(всі двозначні числа)}
```

```
  begin
```

```
    j:=i*n;{для зручності задаємо змінну j– добуток двозначного числа и заданого нами однозначного}
```

```
    a:= i div 10 + i mod 10;{сума цифр двозначного числа}
```

```
    while j<>0 do{сума цифр числа j(в якому кількість цифр може бути більше за два)}
```

```
      begin
```

```
        b:=b+(j mod 10);
```

```
        j:=j div 10;
```

```
      end;
```

```
  if a=b then k:=k+1; {якщо суми цифр двох чисел однакові то лічильник збільшуємо на 1}
```

```
  end;
```

```
  writeln(k);{виводимо кількість чисел}
```

```
end.
```

Перетворення

Візьмемо деяке натуральне число N . Будемо змінювати його наступним чином: якщо число парне, то розділимо його на 2, якщо непарне, додамо 1. Після декількох таких змін ми завжди отримаємо число 1. Наприклад, з числа 11 отримується число 12, потім 6, 3, 4, 2 і, нарешті, 1. Таким чином, для отримання 1 з 11 потрібно виконати 6 перетворень.

Напишіть програму, яка зчитує натуральне число і виводить кількість перетворень даного числа до отримання 1.

Вхідні дані

Число N ($1 \leq N \leq 10^9$).

Вихідні дані

Кількість перетворень.

Вхідні дані

11

Вихідні дані

6

Розв'язок

n- число яке будемо змінювати, k- кількість перетворень. Цикл (**while**) працює до того моменту поки $n \neq 1$. Для визначення парності скористаємось функцією **mod**(остача від ділення). Якщо число парне то ділимо його на 2, в протилежному випадку додаємо 1, при кожному виконанні перетворень лічильник збільшуємо на 1.

```
var k,n:longint;  
begin  
  read(n);  
  k:=0;  
  while n <> 1 do  
  begin  
    if (n mod 2 = 0) then n:=n div 2  
    else n:=n+1;  
    k:=k+1;  
  end;  
  writeln(k);  
end.
```

Гурток хорового співу

У деякому начальному закладі функціонує гурток хорового співу. Початок гуртка завжди відбувається однаково: за сигналом керівника гуртка всі **N** учасників стають у круг і кожен **M**-й для розспівки співає гаму.

Керівник гуртка помітив, що розім'яти горло не завжди вдається всім учасникам гуртка. За заданими **N** і **M** допоможіть йому визначити, чи у черговий раз у розминці приймуть участь всі учасники хору.

Вхідні дані

Вхідні дані складаються з декількох тестових випадків. Кожен тестовий випадок розміщено у окремому рядку і містить два цілих числа **N** і **M**. ($1 \leq N, M \leq 10^3$).

Вихідні дані

Для кожного тестового випадку у окремому рядку виведіть "YES", якщо у розминці приймуть участь всі учасники хору, у протилежному випадку виведіть "NO".

Вхідні дані

4 1
6 3

Вихідні дані

YES
NO

Розв'язок

Рішення задачі зводиться до знаходження найбільшого спільного дільника(НСД). Для цього можна скористатися алгоритмом Евкліда і якщо НСД дорівнює одиниці то у розминці приймуть участь всі учасники хору. В програмі використаний цикл **while not(eof) do**, який зчитує дані поки не кінець файлу.

```
var n,m,nod:int64;  
begin  
  while not(eof) do  
  begin  
    readln(n,m);  
    while n * m <> 0 do {алгоритм Евкліда для знаходження НСД}  
    if n > m then n:= n mod m  
    else m:= m mod n;  
    nod:= n + m;  
    if nod=1 then writeln('YES')  
    else writeln('NO');  
  end;  
end.
```

Цікавий добуток

Визначити всі можливі значення добутку $i*j$, якщо цілочислові значення змінних **i** та **j** змінюються відповідно **i** від **a** до **b** та **j** від **c** до **d** ($1 \leq a, b, c, d \leq 10$).

Вхідні дані

У єдиному рядку 4 числа через пропуск: **a, b, c** та **d**.

Вихідні дані

Єдине число - кількість можливих варіантів добутку.

Вхідні дані

1 10 1 10

Вихідні дані

42

Розв'язок

Для знаходження всіх можливих значень добутку без повторень **i*j** будемо перевіряти кожне число **p** (яке знаходиться в діапазоні від добутку мінімальних значень **i** та **j** до добутку їх максимальних значень) на те, чи можливо його отримати за допомогою даного добутку. Якщо отримати число хоча б один раз, перемноживши **i** та **j**, реально, то значенню **flag** присвоюємо 1. Якщо **flag** дорівнює 1, то збільшуємо лічильник варіантів добутку (**count**).

```
var a,b,c,d,p,ppoch,pkin,i,j,ipoch,ikin,jpoch,jkin,flag,inti,intj,count:integer;
```

```
begin
```

```
  readln(a,b,c,d);
```

```
  if (a<b) then begin ipoch:=a; ikin:=b; end {знаходимо початкове(мінімальне) і кінцеве(максимальне) число проміжку a;b}
```

```
  else begin ipoch:=b; ikin:=a; end;
```

```
  if (c<d) then begin jpoch:=c; jkin:=d; end {знаходимо початкове(мінімальне) і кінцеве(максимальне) число проміжку c;d}
```

```
  else begin jpoch:=d; jkin:=c; end;
```

```
    ppoch:=ipoch*jpoch; {мінімальне значення добутку}
```

```
    pkin:=ikin*jkin; {максимальне значення добутку}
```

```
    count:=0;
```

```
    for p:=ppoch to pkin do {перебір всіх значень добутку}
```

```
      begin
```

```
        flag:=0;
```

```
        for inti:=ipoch to ikin do {перебір всіх можливих варіантів добутку}
```

```
          for intj:=jpoch to jkin do
```

```
            if p=(inti*intj) then flag:=1; {перевірка чи можливо отримати число з проміжку(від мінімального значення до максимального) перемноживши inti*intj}
```

```
            if flag=1 then count:=count+1; {якщо знайшлось хоча б одне число то збільшуємо лічильник}
```

```
          end;
```

```
        writeln(count)
```

```
      end.
```