



CHAPTER 11

A Simple Program Whose Proof Isn't

NHÓM 10_DASA230179_09
GIẢNG VIÊN HƯỚNG DẪN: THS. HUỲNH XUÂN PHỤNG

THÀNH VIÊN NHÓM 10



Hồ Khánh Đăng
21110876



Trần Tuấn Anh
21110369



Lê Hữu Huy
21110471

NỘI DUNG TỔNG QUAN CỦA BÁO CÁO

Giới thiệu đề tài

Giới thiệu tổng quan về đề tài, chương trình TeX và vấn đề

Nội dung chính của đề tài

Chuyển đổi qua lại giữa phân số thập phân và chấm động nhị phân

Ứng dụng của đề tài

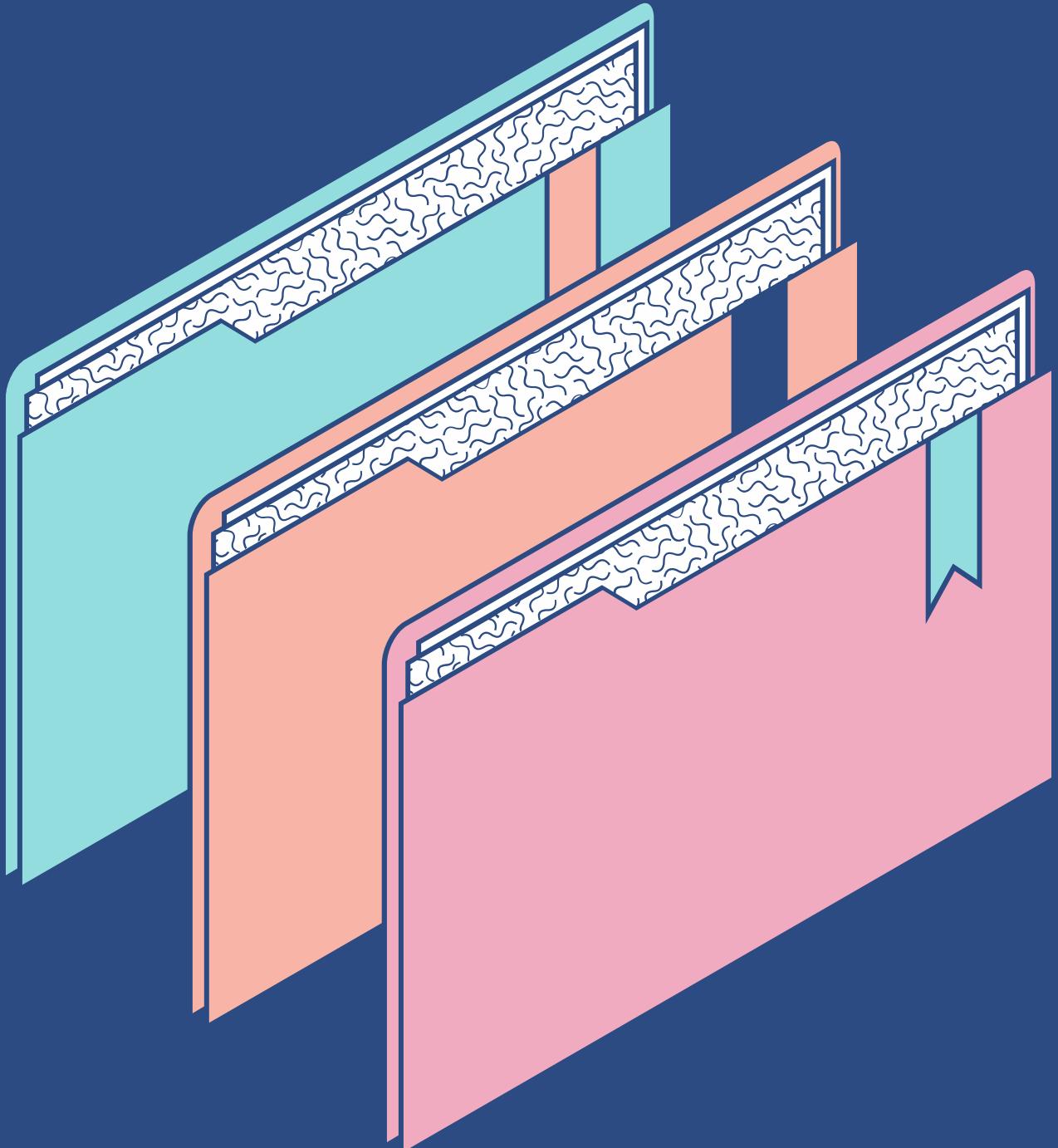
Ứng dụng của chương trình vào thực tế



GIỚI THIỆU ĐỀ TÀI

NỘI DUNG CHÍNH:

- Cách chuyển qua lại giữa phân số thập phân và số chấm động nhị phân.
- Cách chứng minh tính đúng đắn của 1 chương trình đơn giản khi chuyển qua lại giữa phân số thập phân và số chấm nhị phân

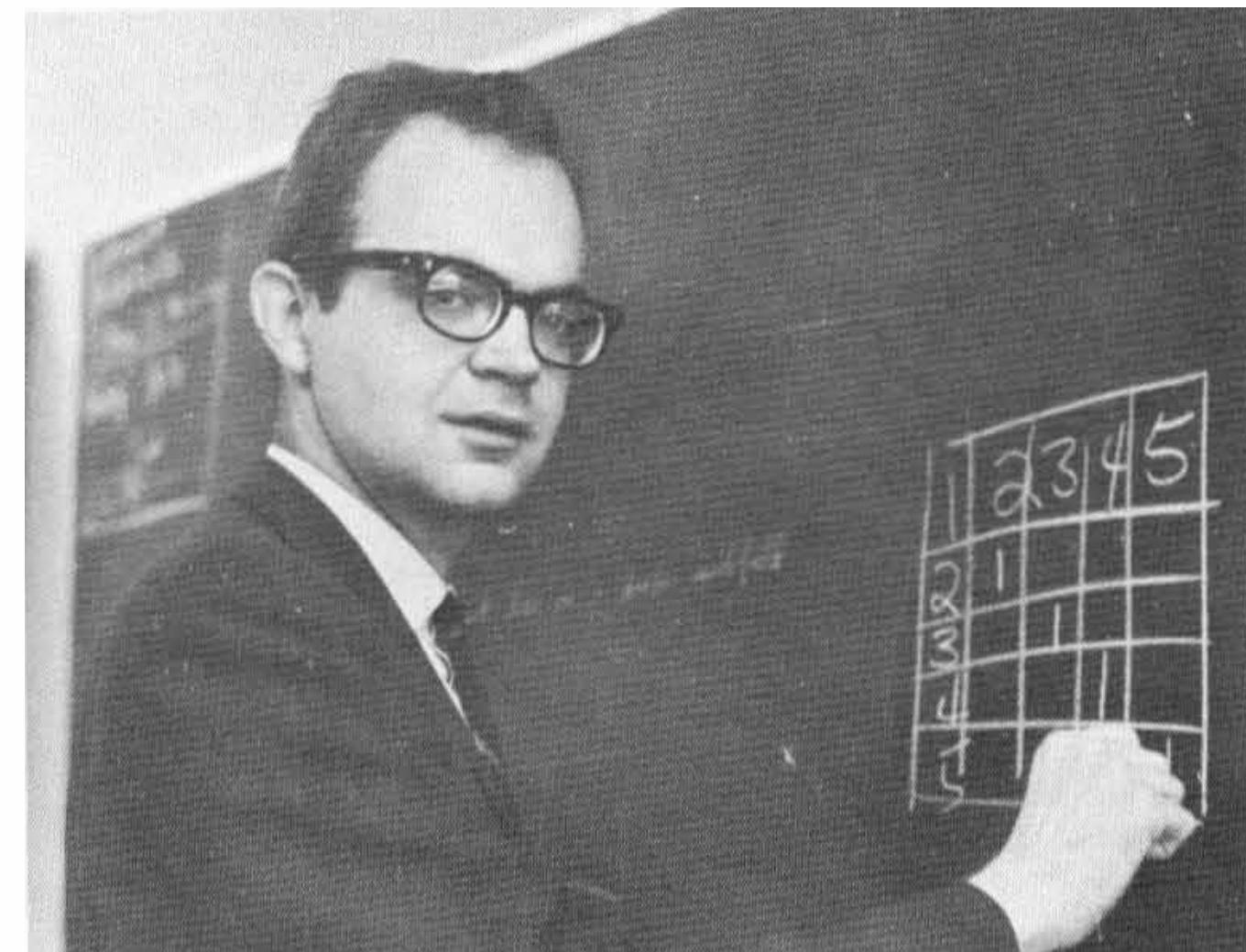


Giới thiệu về chương trình Tex

-Chương trình là một hệ thống sắp chữ được viết bởi Donald Knuth và giới thiệu lần đầu vào năm 1978.

-TeX được thiết kế với hai mục đích chính:

- Cho phép bất kì ai cũng có thể tạo ra những văn bản chất lượng cao với ít công sức nhất
- Cung cấp một hệ thống sắp chữ cho ra cùng một kết quả trên mọi máy tính.

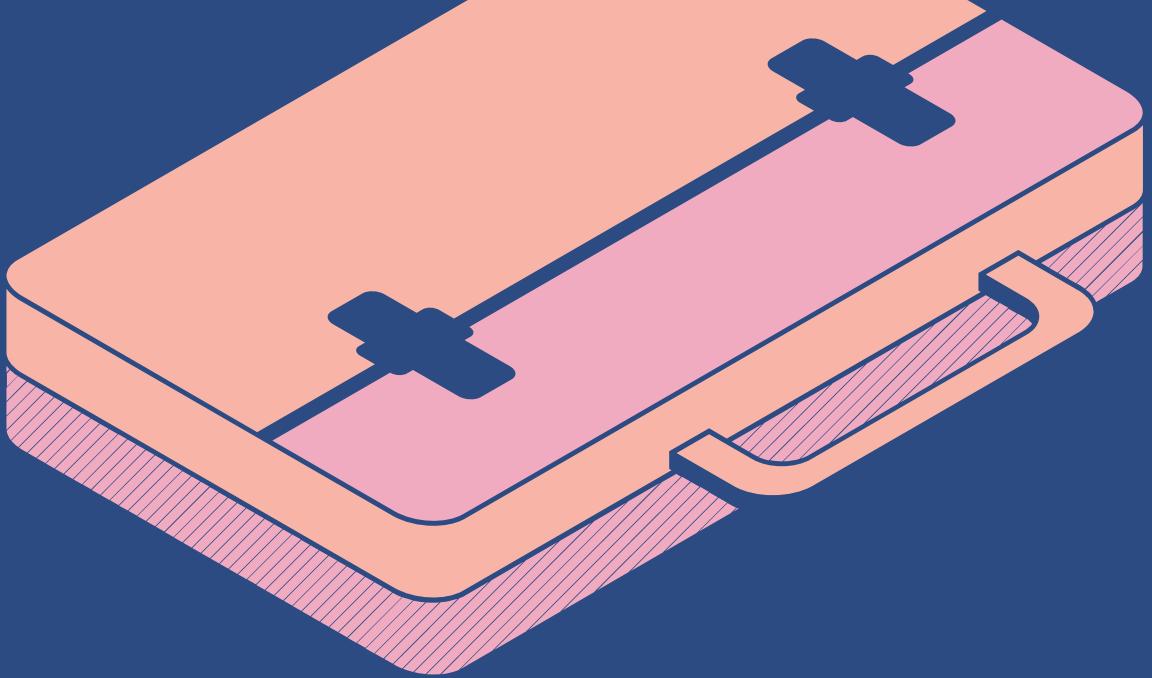


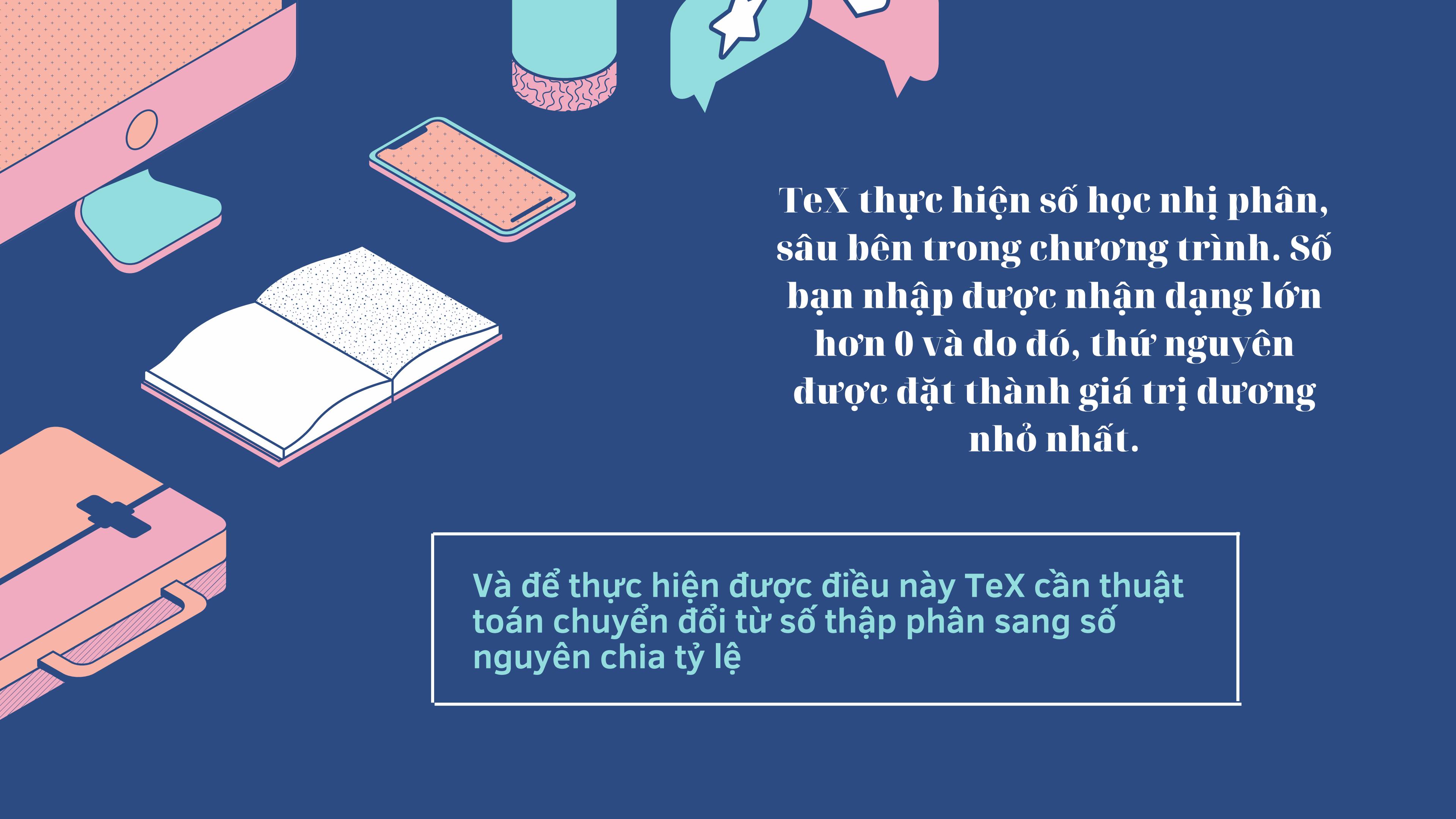
Chương trình TeX này hoạt động dựa trên bộ số của 2^{16} .

Khi tác giả đang viết chương trình TEX ông cần 1 chương trình đơn giản để giải quyết các vấn đề nhỏ trong chương trình TEX như chuyển đổi từ phân số thập phân sang số nhị phân chấm động và ngược lại.

Tác giả đã tìm ra giải pháp cho một trong các vấn đề đó hóa ra dễ vì nó ngắn và chỉ là một đoạn code đơn giản. Nhưng khó trong vấn đề chứng minh và giải thích tính đúng đắn của vấn đề.

CÁC VẤN ĐỀ CỦA ĐỀ TÀI

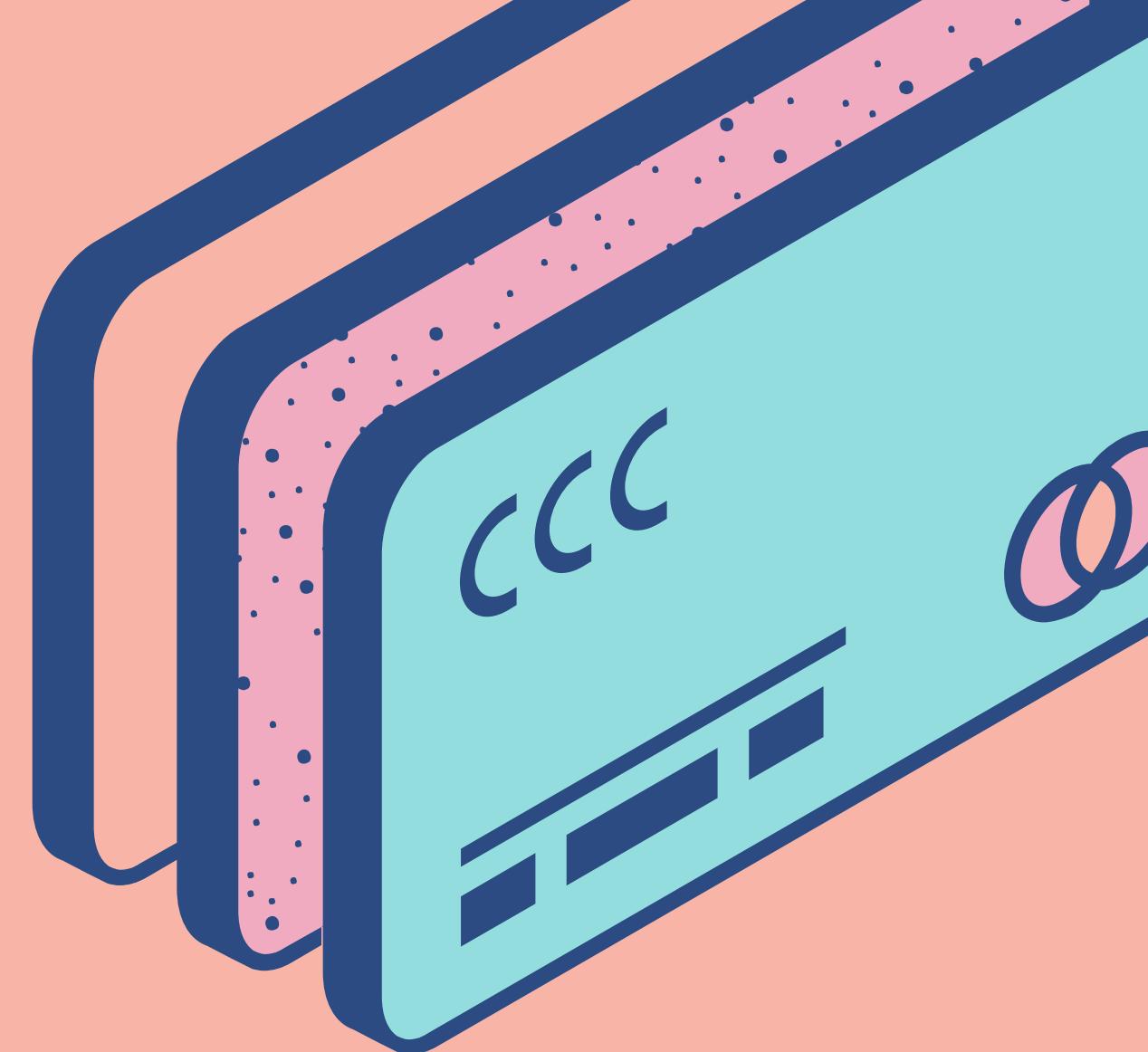




**TeX thực hiện số học nhị phân,
sâu bên trong chương trình. Số
bạn nhập được nhận dạng lớn
hơn 0 và do đó, thứ nguyên
được đặt thành giá trị dương
nhỏ nhất.**

**Và để thực hiện được điều này TeX cần thuật
toán chuyển đổi từ số thập phân sang số
nguyên chia tỷ lệ**

NỘI DUNG CHÍNH CỦA BÁO CÁO



1. Converting Decimal Fraction to Fixed-Point Binary

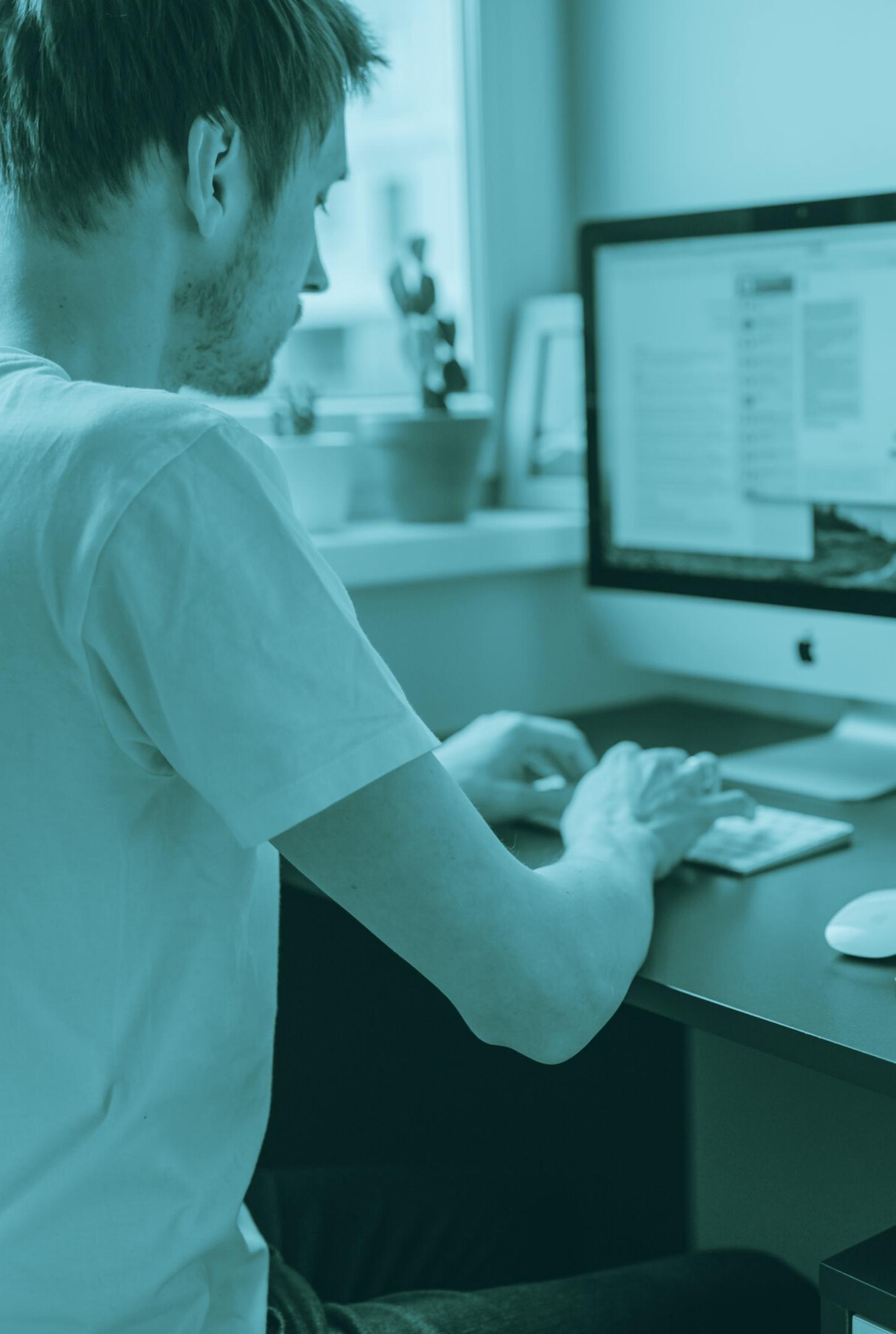
Chuyển đổi từ phân số thập phân sang số nhị phân chấm động

2. Converting Fixed-Point Binary to Decimal Fraction

Chuyển đổi ngược lại và chứng minh thuật toán

3. Extended algorithm

Mở rộng thuật toán



1. Converting Decimal Fraction to Fixed-Point Binary

Nội dung

-Chuyển đổi từ phân số thập phân sang số nhị phân chấm động

Lý do cần chuyển đổi:

- Vì chương trình TEX chỉ hoạt động trên bộ số của 2-16 .
- Đầu vào của chương trình TeX là một số thập phân.
- >Nên cần 1 chương trình con biên dịch từ số thập phân ấy sang số chấm động nhị phân.

PHÂN TÍCH THUẬT TOÁN P1

INPUT

- Đầu vào là: .d₁d₂d₃...d_k.

VD: 2.25 PHẦN NGUYÊN LÀ 2 VÀ PHẦN THỰC LÀ .25

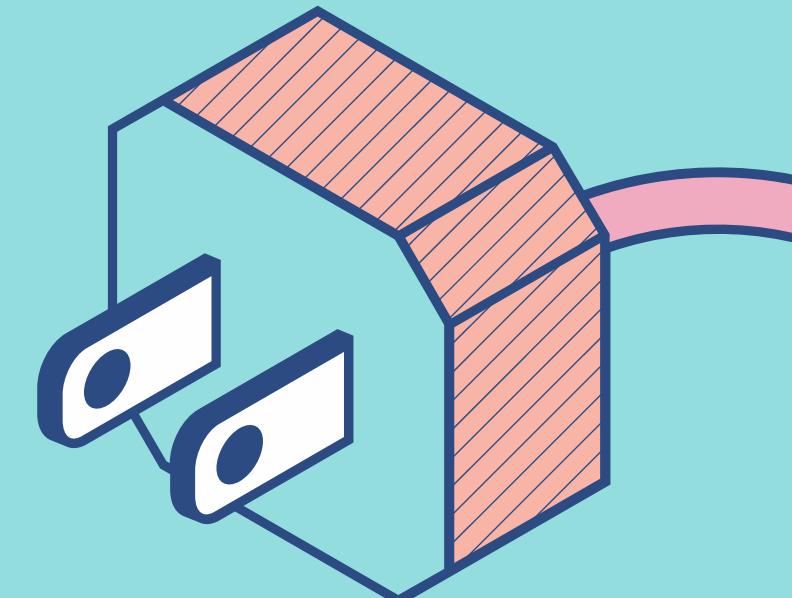
CÔNG THỨC TỔNG QUÁT

$$n = \left\lfloor 2^{16} \sum_{j=1}^k d_j / 10^j + 1/2 \right\rfloor.$$

VD .25 thì n = [16384.5] = 16384

OUTPUT

- Đầu ra là số chấm động nhị phân n



Chạy thuật toán dựa theo chương trình trên TeX

- Ở đây tác giả dựa trên chương trình TeX để viết thuật toán.
- Chương trình TeX có khả năng lưu trữ được tối 17 số sau dấu chấm thập phân từ.
 $.d_1d_2d_3\dots d_{17}.$

Công thức chạy trên TeX

$$n = \left\lfloor 2^{16} \sum_{j=1}^{17} d_j / 10^j + 1/2 \right\rfloor.$$

CÀI ĐẶT CHƯƠNG TRÌNH BẰNG C++ P1

```
20
21 long long fixedPointBinary(int D[],int k)
22 {
23     int l=min(k,17);
24     double m=0;
25     do{
26         m = (131072*D[l-1]+m)/10;
27         l--;
28     }while(l!=0);
29     long long n=(m+1)/2;
30     return n;
31 }
32
```

$$n = \left[2^{16} \sum_{j=1}^{17} d_j / 10^j + 1/2 \right].$$

Input , Output

input: **d1d2...d17**

output: giá trị n

Code

Quy đồng 2

It is convenient to compute n by writing

$$n = \left[\frac{m_0 + 1}{2} \right], \quad m_l = \left[2^{17} \sum_{j=l+1}^k d_j / 10^{j-l} \right].$$

```

20
21 long long fixedPointBinary(int D[],int k)
22 {
23     int l=min(k,17);
24     double m=0;
25     do{
26         m = (131072*D[l-1]+m)/10;
27         l--;
28     }while(l!=0);
29     long long n=(m+1)/2;
30     return n;
31 }
32

```

$$n = \left\lfloor 2^{16} \sum_{j=1}^{17} d_j / 10^j + 1/2 \right\rfloor.$$

It is convenient to compute n by writing

$$n = \left\lfloor \frac{m_0 + 1}{2} \right\rfloor, \quad m_l = \left\lfloor 2^{17} \sum_{j=l+1}^k d_j / 10^{j-l} \right\rfloor.$$

Các bước trong thuật toán

Tùy công thức

B1: Gắn biến l theo k và tối thiểu của l có thể nhận là 17 số nên $l=\min(k,17)$; gắn $m=0$

B2: Thực hiện vòng lặp tính m theo công thức m_l

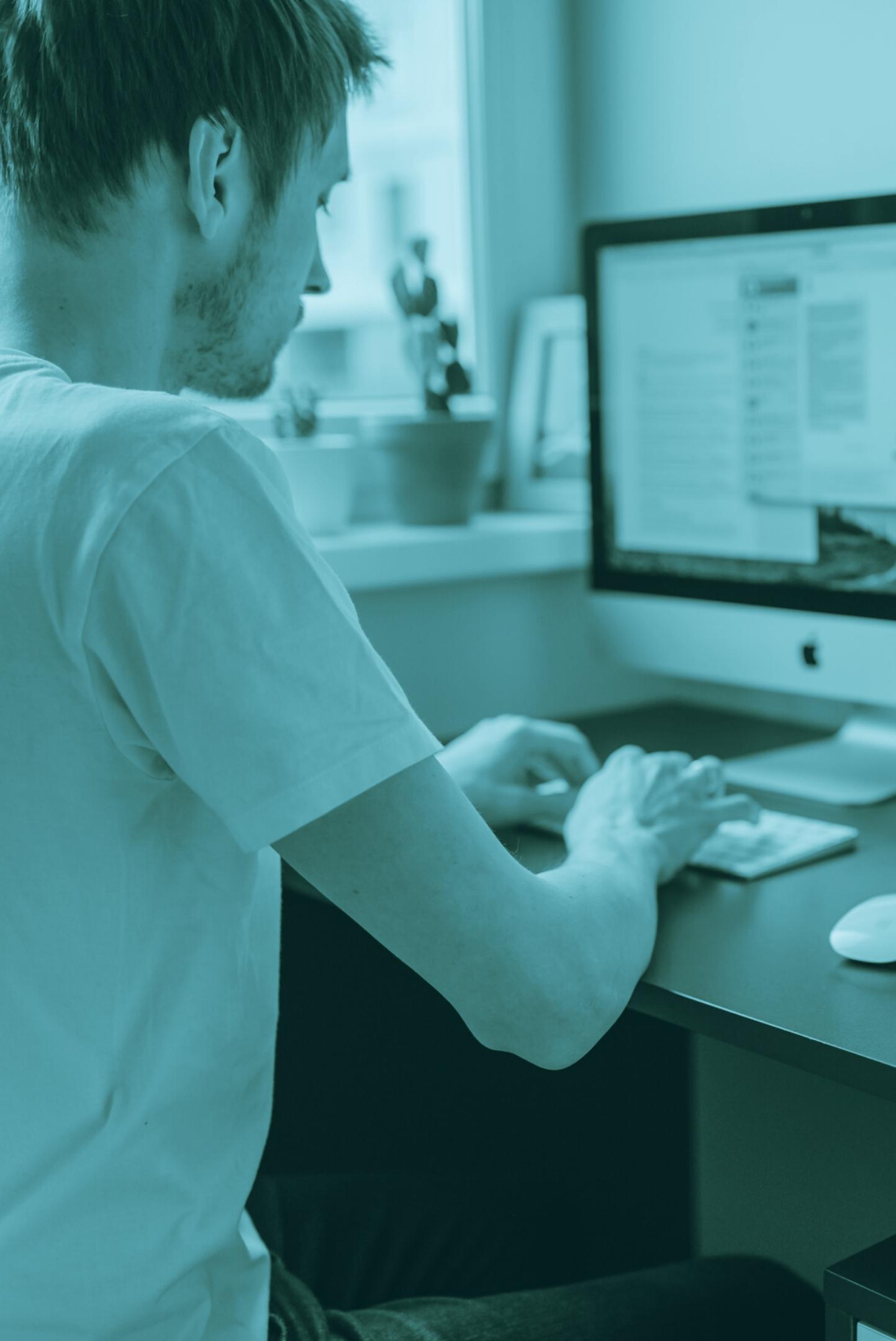
B3: Tính n sau khi tính được m theo công thức $(m+1)/2$.

Ví dụ chạy tay:

Dầu vào: là số sau chấm phân số .25

Dầu ra: $n=16384$

		0.25		
	d1=2	d2=5		
B1	l=2	m=0		
B2			Chạy vòng lặp do while(tới khi l=0)	
	l=1	m=65536	giảm l--;	
	l=0	m=32768	giảm l--;	
B3		n=16384		



2. Converting the Other Way

Nội dung

Chuyển đổi ngược lại so với phần 1

Tác giả đưa ra thuật toán giải quyết vấn đề sau

Cho

$$n \in [0, 2^{16}]$$

Tìm các chữ số
phần thập phân

.d₁d₂d₃...d_k.

-Ở thuật toán sẽ tìm 5 chữ số của số thập phân $n/2^{16}$ (tức là $k=5$)

Vì sao lại chọn k=5 ?

PHÂN TÍCH THUẬT TOÁN P2

INPUT

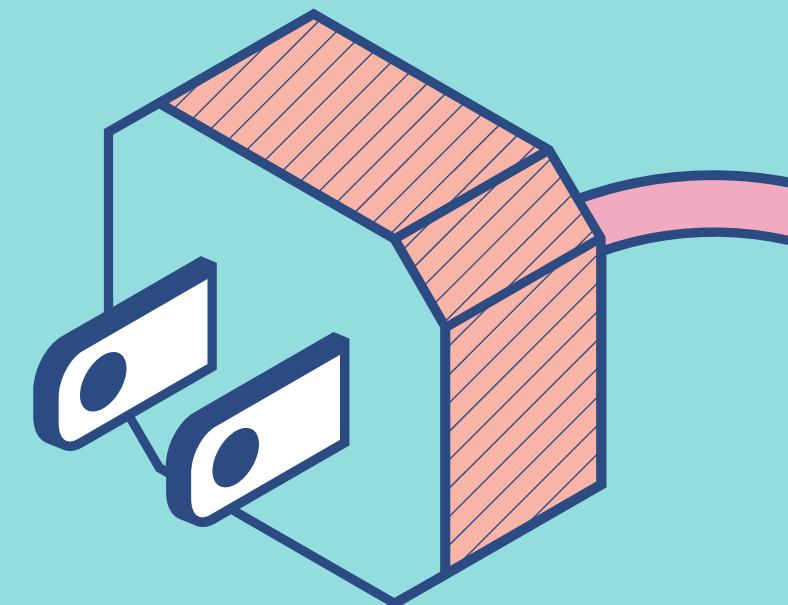
-Đầu vào là: .d1d2d3...dk.

OUTPUT

-Đầu ra là số chấm động nhị phân n

CHƯƠNG TRÌNH TỔNG QUÁT

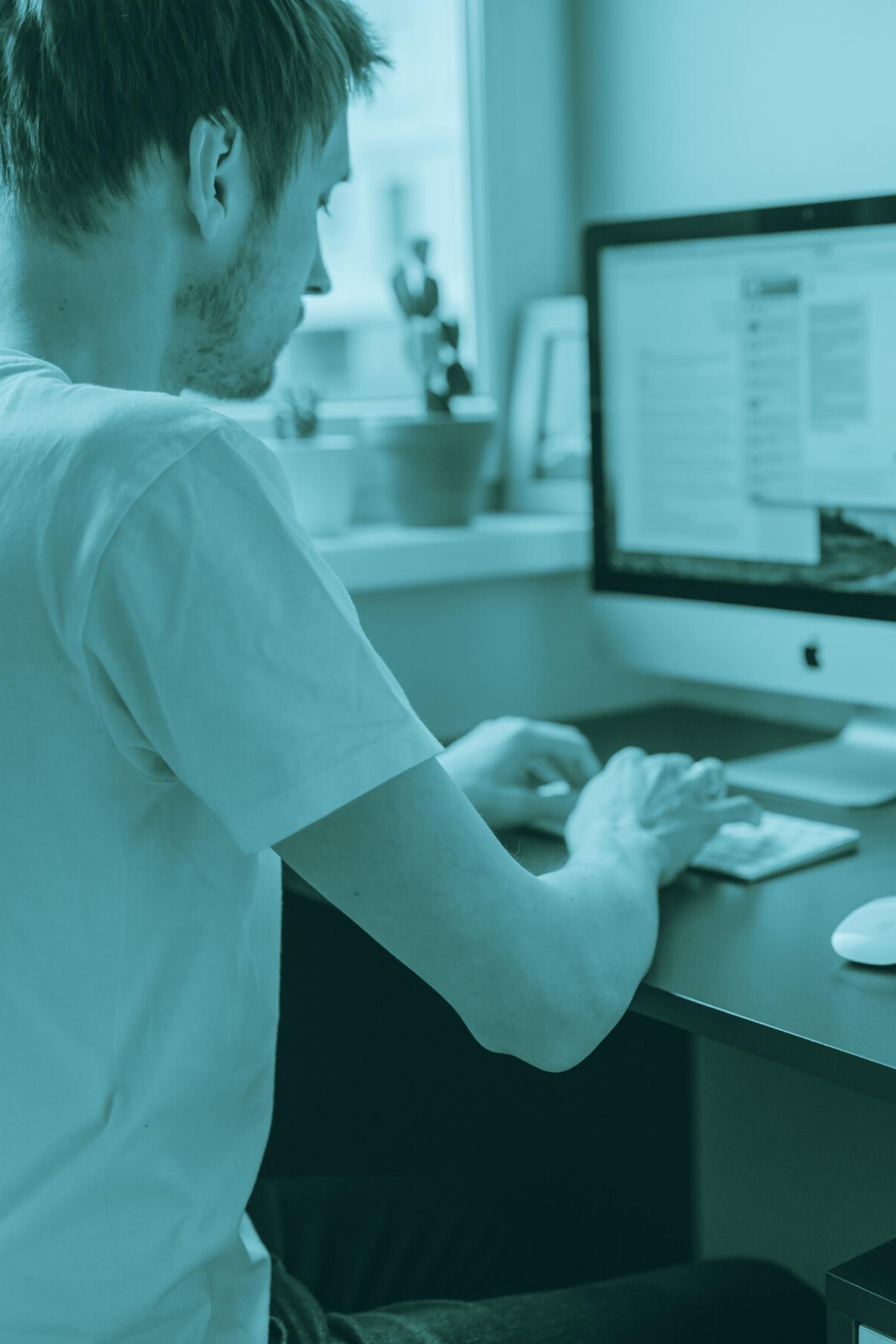
```
P2:  j := 0;  s := 10 * n + 5;  t := 10;  
repeat if t > 65536 then s := s + 32768 - (t div 2);  
    j := j + 1;  d[j] := s div 65536;  
    s := 10 * (s mod 65536);  t := 10 * t;  
until s ≤ t;  
k := j.
```



CÀI ĐẶT CHƯƠNG TRÌNH BẰNG C++ P2

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4
5 int gcd(int a,int b)
6 {
7     if (b==0) return a;
8     return gcd(b,a%b);
9 }
10
```

```
11 void decimalFraction()
12 {
13     int n; cin>>n;
14     int j=0;
15     int s = 10*n;
16     int t = 10;
17     int d[100];
18     while (s>=t)
19     {
20         if (t>65536) s = s + (65536/2);
21         j++; d[j] = s/65536;
22         s = 10*(s%65536); t*=10;
23     }
24     int k = j;
25     cout<<k<<endl;
26     for (int i=1;i<=k;i++) cout<<d[i]<<" ";
27 }
28
```



3. Germ of a proof

Vấn đề

Tại sao thuật toán chỉ tìm nhiều nhất 5 chữ số của $n/2^{16}$?.

**Tác giả chứng minh được rằng số thập phân mà
tác giả tìm được là tối ưu nhất**

Gọi số thập phân của tác giả là

$d_1d_2d_3\dots d_k \ (k \leq 5)$

Chứng minh phản chứng

Giả sử có số: $d'_1 d'_2 \dots d'_k$ với $.K' < K$ và $d'_1 d'_2 \dots d'_k < d_1 d_2 \dots d_k$

Đặt $x = d'_1 d'_2 \dots d'_k$

Đặt $x' = d_1 d_2 \dots d_k$

Số thỏa mãn gần đúng nhất với $n \backslash 2^{16}$ là số:

$$\left| x - \frac{n}{2^{16}} \right| \leq 2^{-17} \quad (x \approx 2^{16})$$

$$\left| x - \frac{n}{2^{16}} \right| \leq 2^{-17}$$

Nếu ta có

$$\left| x' - \frac{n}{2^{16}} \right| \leq 2^{-17}$$

Ta có:

$$x - x' \geq 10^{-k'} \text{ vì } (k' < k)$$

Xét

$$\left| x - \frac{n}{2^{16}} \right| + \left| x' - \frac{n}{2^{16}} \right|$$

$$= \left| x - \frac{n}{2^{16}} \right| + \left| \frac{n}{2^{16}} - x' \right|$$

Ta có BĐT đă học:

$$|a| + |b| \geq |a + b|$$

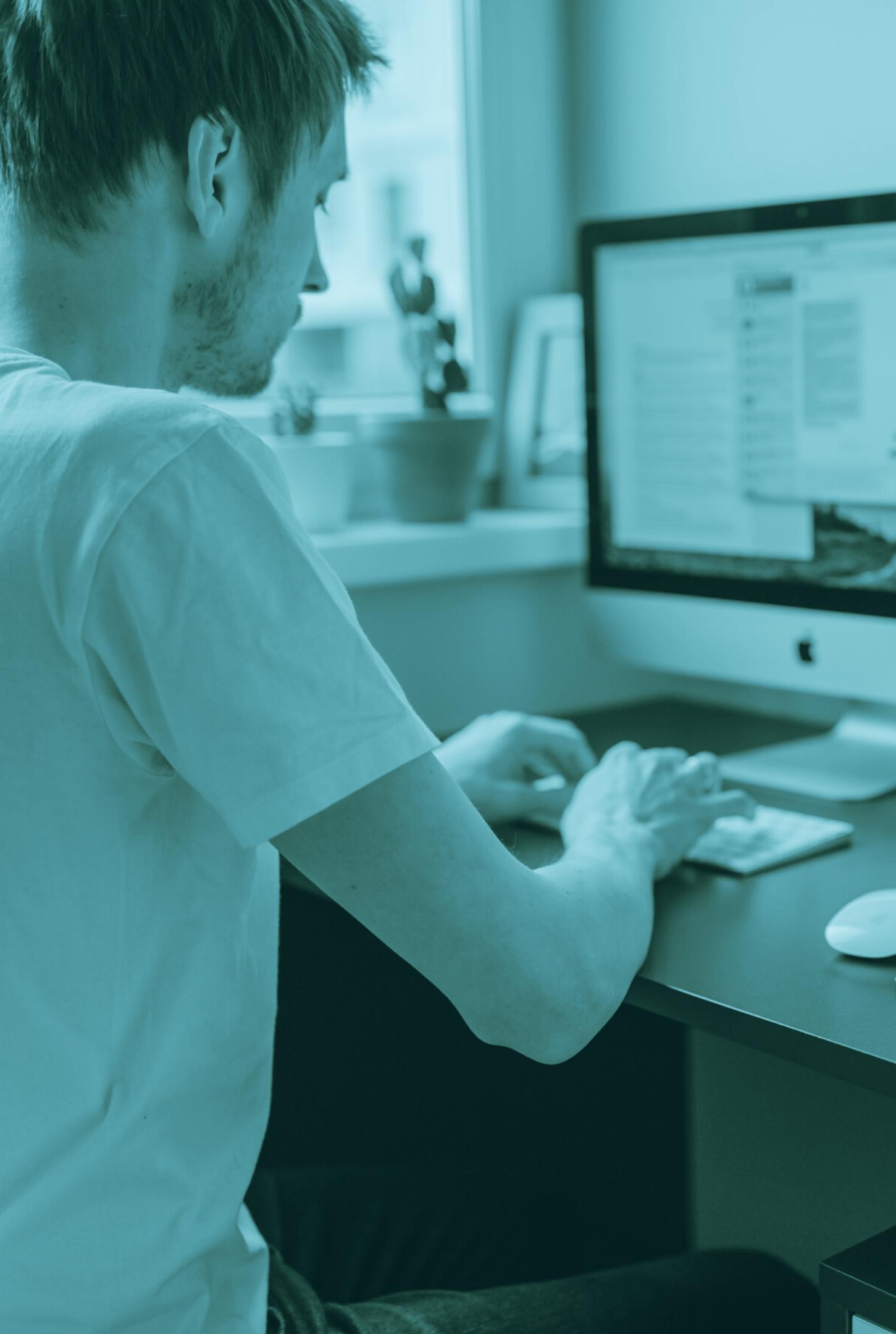
$$\Rightarrow \left| x - \frac{n}{2^{16}} \right| + \left| \frac{n}{2^{16}} - x' \right| \geq \left| x - \frac{n}{2^{16}} + \frac{n}{2^{16}} - x' \right| = |\mathbf{x} - \mathbf{x}'| = 10^{-k'}$$

Suy ra $10^{-k'} \leq 2^{-17} + 2^{-17} = 2^{-16}$ $\Leftrightarrow k' \geq 5 \geq k$

Mâu thuẫn với giải sử ban đầu là $k' < k \Rightarrow$ giả sử sai

Vậy thập phân tối ưu nhất biểu diễn gần đúng $n/2^{16}$

Hay nói cách khác là không có số thập phân nào ngắn hơn $d_1d_2\dots d_k$ thỏa
tính chất $|d_1\dots d_n - \frac{n}{2^{16}}| \leq 2^{-17}$



4. Completion of the Proof

-**Thuật toán của tác giả lấy nhiều nhất 5 chữ số thập phân nên điều là $s \leq t$**
Khi $t = 10^6$ tức là thuật toán chạy được 5 vòng lặp, ta thấy: t lúc này sẽ lớn hơn s. Bởi vì : $mã(s) = 10(2^{16}-1) = 655350 < 10^6 = t$
Nên khi xong vòng lặp thứ 5 thì thuật toán chắc chắn dừng lại

-Ta giả thích ý nghĩa câu lệnh

```
if t > 65536 then s := s + 32768 - (t div 2)
```

Ta xuất phát từ dòng lệnh đầu tiên

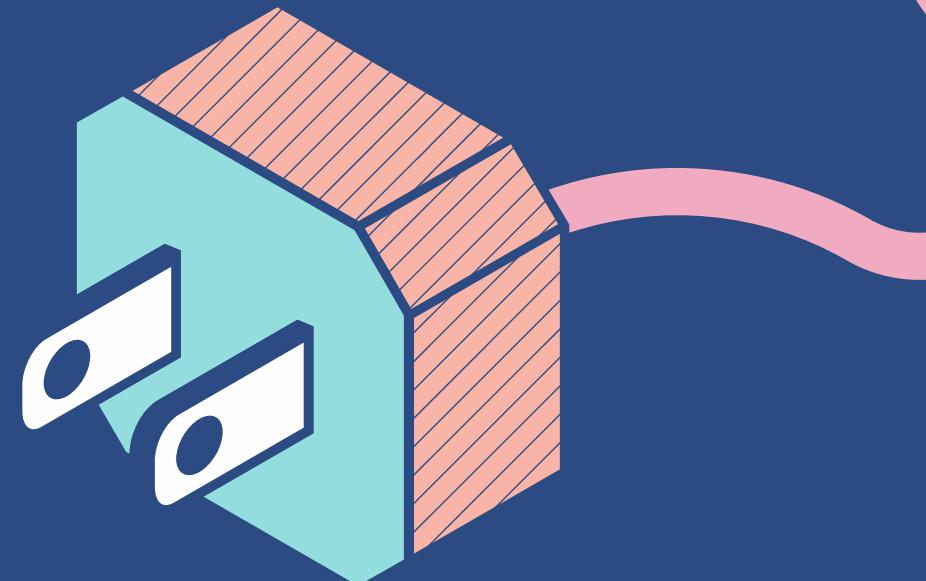
+Tác giả đặt $s = 10n+5$

+Ta chú ý đến số 5, ta thấy sau mỗi vòng lặp thì 5 sẽ nhân 10

Khi ở vòng lặp thứ 6 thì 5 sẽ tăng thành $5 \cdot 10^5 > 65536$

Như vậy từ vòng lặp thứ 5 trở về trước thì yếu tố 5
không làm ảnh hưởng tới kết quả

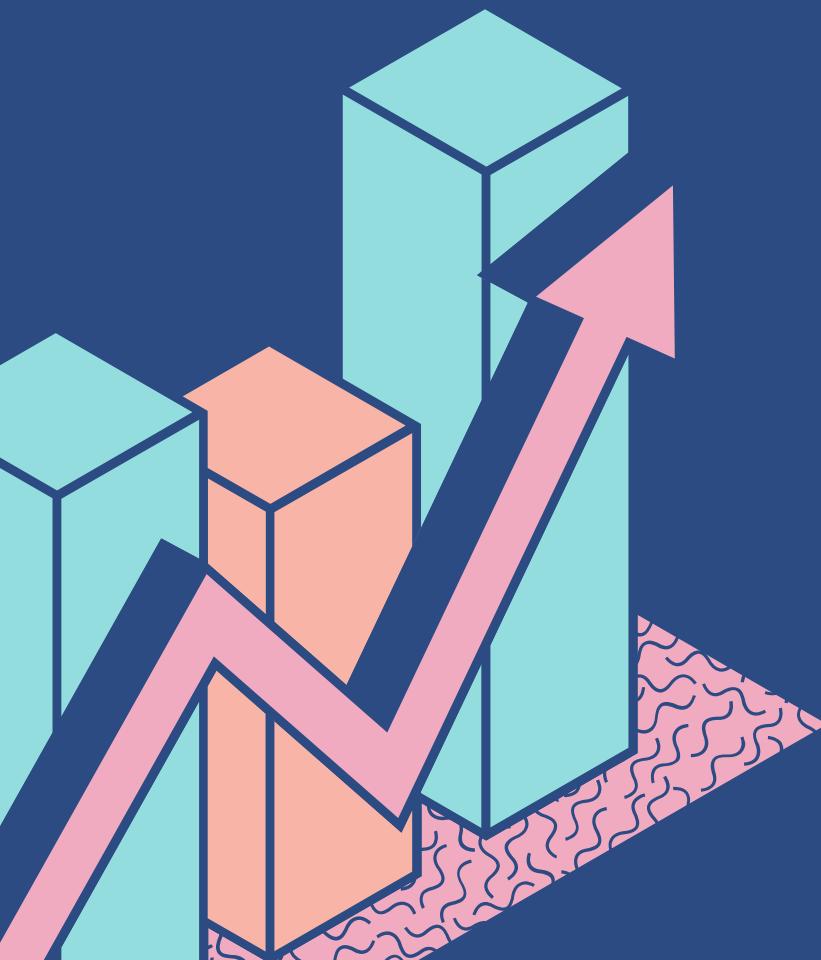
5. Closing remarks



-Thuật toán P3 tương tự như P2 nhưng có vài sự thay đổi

Kết quả $.d_1 \dots d_k$ là một biểu diễn tối ưu tùy thuộc vào điều kiện :

$$\frac{n - 1/2}{m} \leq .d_1 \dots d_k < \frac{n + 1/2}{m}.$$



PHÂN TÍCH THUẬT TOÁN P3

INPUT

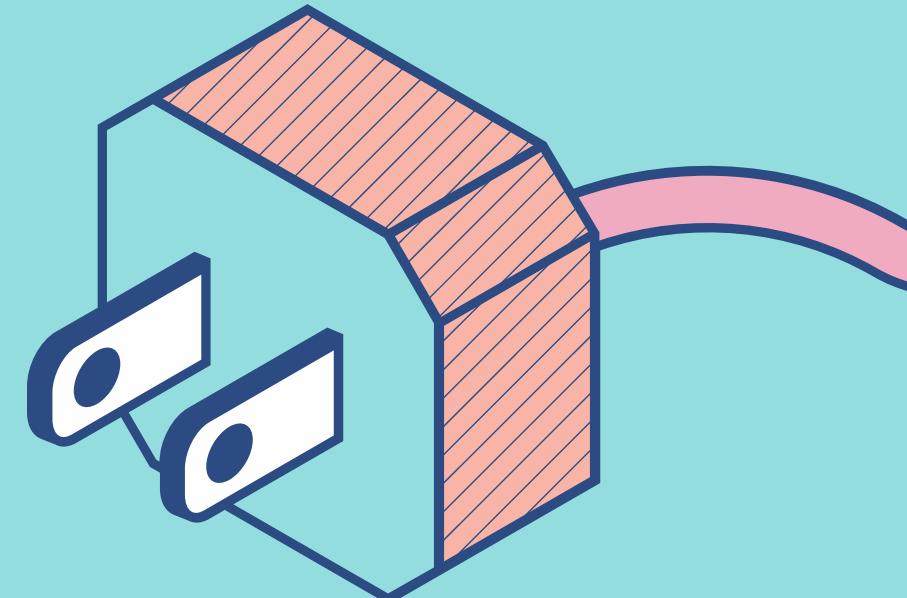
-Số nguyên n,m là số thập phân n/m

OUTPUT

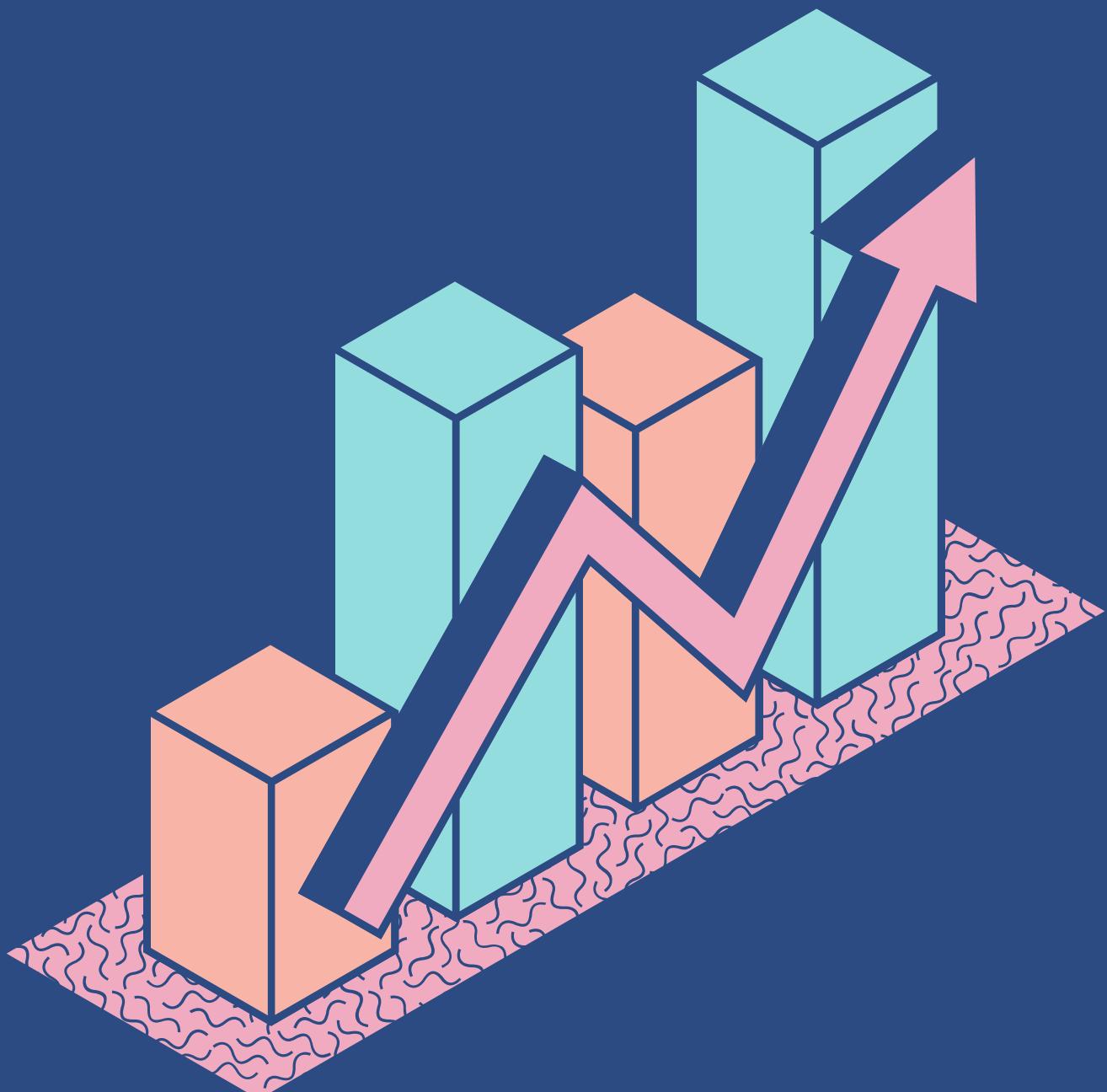
-Số thập phân dưới dạng .d1d2...dk sao cho $n \cdot 2^{-16} = .d1d2...dk$

CHƯƠNG TRÌNH TỔNG QUÁT

```
j := 0; s := 10 * n + 4; t := 10;  
repeat if t > m then s := s + 1 + (m div 2) - (t div 2);  
    j := j + 1; d[j] := s div m;  
    s := 10 * (s mod m) + 9; t := 10 * t;  
until s < t;  
k := j.
```



ỨNG DỤNG THUẬT TOÁN



Ứng dụng cho ngành toán học

Chuyển số thập phân có dấu sang số nhị phân đơn giản nhất

Chương trình tex

Là tiền thân của chương trình LaTex sau này

Các ngôn ngữ như c++,c#...hay tin học văn phòng

Sử dụng cho các hàm về làm tròn

Lập trình máy tính casio

Ứng dụng rộng rãi cho lập trình máy tính casio

Các app khác

Ứng dụng rộng rãi trong các app để hiển thị ra kết quả vừa ngắn và xấp xỉ gần đúng nhất



**Thank You
for listening!**

