Complete Non-Computable Function Guide: Solve ANY Exercise

Universal Truth: Only Two Methods Exist

Every non-computable function exercise uses exactly one of these:

1. **Diagonalization**: Build $f(x) \neq \varphi_x(x)$ for all x
2. **χ_k Method**: Use characteristic function of halting set

Phase 1: Instant Pattern Recognition (30 seconds)

Read the question and classify:

Type A: Build total non-computable function with [constraints] → Use Diagonalization

Type B: Function with infinite equality property (f(x) = f(x+1) infinitely, etc.) → Use χ_k

Type C: Prove given function is non-computable → Use χ_k Reduction

Type D: Composition/arithmetic properties → Use χ_k or Modified Diagonalization

Phase 2: Mechanical Execution

METHOD 1: DIAGONALIZATION (Most Common)

Universal Template

javascript

```javascript
f(x) = {
    modify(φₓ(x))     if φₓ(x) ↓
    constant          if φₓ(x) ↑
}
```

Step-by-Step Process:

Step 1: Choose modification function

- Basic case: φ�x(x) + 1
- For constraints: map to required codomain

Step 2: Choose constant

- Default: 0
- For constraints: value in required codomain

Step 3: Verify $f(x) \neq \varphi_x(x)$ always

- Case 1: $\varphi_x(x)\downarrow \rightarrow f(x)$ = modified $\neq \varphi_x(x)$ ✓
- Case 2: $\varphi_x(x)\uparrow \rightarrow f(x)$ = constant $\neq \uparrow$ ✓

Complete Examples for All Constraint Types:

A1. Basic (no constraints)

```javascript
f(x) = {
    ϕₓ(x) + 1    if ϕₓ(x) ↓
    0            if ϕₓ(x) ↑
}
```

A2. Codomain = {0,1}

```javascript
f(x) = {
    sg(ϕₓ(x))    if ϕₓ(x) ↓
    0            if ϕₓ(x) ↑
}
```

Note: sg(y) = 0 if y=0, 1 if y>0

A3. Image = ℕ{0} (exclude zero)

```javascript
f(x) = {
    ϕₓ(x) + 1    if ϕₓ(x) ↓
    1            if ϕₓ(x) ↑
}
```

A4. Image = $\{2^n \mid n \in \mathbb{N}\}$ (powers of 2)

```javascript
f(x) = {
    2^(ϕₓ(x)+1)  if ϕₓ(x) ↓ ≠ 0
    2            if ϕₓ(x) ↑ OR ϕₓ(x) = 0
}
```

A5. Image = Primes

```javascript
f(x) = {
    next_prime_after(ϕₓ(x))  if ϕₓ(x) ↓
    2                        if ϕₓ(x) ↑
}
```

A6. f(x) = x for infinitely many x

```javascript
f(x) = {
    ϕₓ(x) + 1    if x ∈ Wₓ
    x            if x ∉ Wₓ
}
```

Key: ∅ has infinitely many indices → $x \notin W_x$ infinitely often

A7. f returns 0 when x even

```javascript
f(x) = {
    0                         if x even
    ϕ_{(x-1)/2}(x) + 1       if x odd ∧ ϕ_{(x-1)/2}(x) ↓
    0                         if x odd ∧ ϕ_{(x-1)/2}(x) ↑
}
```

A8. Domain ∩ Codomain = ∅

```javascript
f(x) = {
    2·χₖ(⌊x/2⌋)    if x odd
    ↑               if x even
}
```

Domain = odd numbers, Codomain = {0,2}

A9. Finite codomain with constraints

```javascript
f(x) = {
    ϕₓ(x) mod k + 1  if ϕₓ(x) ↓
    1                if ϕₓ(x) ↑
}
```

Ensures codomain ⊆ {1,2,...,k}

METHOD 2: χ_k (Characteristic Function of K)

When to Use χ_k:

- "f(x) = f(x+1) infinitely often"
- "Prove function g is non-computable"
- "{0,1} codomain with specific properties"
- "Almost total" functions

χ_k Definition:

```javascript
χₖ(x) = {
    1    if ϕₓ(x) ↓ (x ∈ K)
    0    if ϕₓ(x) ↑ (x ∉ K)
}
```

Complete Examples:

B1. f(x) = f(x+1) infinitely often Answer: f = χ_k

Proof: If $\{x \mid \chi_k(x) = \chi_k(x+1)\}$ were finite with max d, then:

- For x > d: $\chi_k(x+1) = sg(\chi_k(x))$
- Define χ_k by primitive recursion → computable → contradiction!

B2. Prove g(x) = {2x+1 if $\varphi_x(x)$ ↓; 2x-1 if $\varphi_x(x)$ ↑} non-computable

```javascript
χₖ(x) = sg(|g(x) - 2x|)
```

If g computable → χ_k computable → contradiction

B3. Almost total function

```javascript
f(x) = {
    0           if x ∈ Wₓ
    ϕₓ(x) + 1   if x ∉ Wₓ
}
```

If f computable & almost total → can compute χ_k → contradiction

METHOD 3: COMPOSITION/ARITHMETIC PROPERTIES

C1. f + g computable but f,g non-computable

Answer: $f = \chi_k$, $g = \chi_{\bar{k}}$ Then f + g = 1 (constant, computable)

C2. f ∘ g computable but g non-computable

Example: $g = \chi_k$, f(x) = 0 for all x Then f ∘ g = 0 (constant, computable)

C3. sg ∘ f computable but f non-computable

If sg ∘ f computable, then f cannot have finite codomain containing 0,
use modified diagonalization ensuring f(x) ≠ 0 infinitely often

UNIVERSAL DECISION FLOWCHART

```
Step 1: What does the question ask?

├─ "Build total non-computable function" → DIAGONALIZATION
│   ├─ With codomain constraints? → Use appropriate template A2-A9
│   ├─ With arithmetic property? → Use template A6-A7
│   └─ Basic case? → Use template A1
│
├─ "f(x) = f(x+k) infinitely" → χₖ (template B1)
│
├─ "Prove function non-computable" → χₖ REDUCTION (template B2)
│
├─ "Composition property" → χₖ or SPECIAL (templates C1-C3)
│
└─ "Almost total/partial function" → χₖ (template B3)
```

VERIFICATION CHECKLIST (Mandatory!)

For Diagonalization:

- [ ] Function is total (both $\varphi_x(x)\downarrow$ and $\varphi_x(x)\uparrow$ cases covered)
- [ ] $f(x) \neq \varphi_x(x)$ for ALL x (this is THE crucial step!)
- [ ] Case $\varphi_x(x)\downarrow$: f(x) = [modified] ≠ $\varphi_x(x)$
- [ ] Case $\varphi_x(x)\uparrow$: f(x) = [constant] ≠ ↑
- [ ] All constraints satisfied (codomain, image, etc.)
- [ ] Clear statement: "f is non-computable"

For χ_k Method:

- [ ] Correctly stated $\chi_k(x) = \{1 \text{ if } \varphi_x(x)\downarrow; 0 \text{ if } \varphi_x(x)\uparrow\}$
- [ ] Proper application (direct use OR reduction)
- [ ] Clear contradiction argument
- [ ] Conclusion: "therefore non-computable"

EXAM EXECUTION STRATEGY (10 minutes)

Minutes 0-1: Pattern Recognition

Use flowchart above → identify method instantly

Minutes 1-2: Template Selection

- Diagonalization → choose template A1-A9
- χ_k → choose template B1-B3
- Special → choose template C1-C3

Minutes 2-8: Mechanical Execution

Copy template exactly, fill in specifics

Minutes 8-10: Verification

Use checklist above - this gets you full points!

EMERGENCY PROTOCOL

If completely stuck:

1. **Default to basic diagonalization**: $f(x) = \{\varphi_x(x) + 1 \text{ if } \varphi_x(x) \downarrow; 0 \text{ if } \varphi_x(x) \uparrow\}$
2. **Verify $f(x) \neq \varphi_x(x)$ always**
3. **Adapt for any constraints mentioned**
4. **Write verification explicitly**

If diagonalization doesn't fit:

1. **Try χ_k direct application**
2. **Write $\chi_k(x)$ = expression involving given function**
3. **Conclude by contradiction**

KEY FORMULAS FOR EXAM

Basic Diagonalization: $f(x) = \{\varphi_x(x) + 1 \text{ if } \varphi_x(x) \downarrow; 0 \text{ if } \varphi_x(x) \uparrow\}$ **χ_k**: $\chi_k(x) = \{1 \text{ if } \varphi_x(x) \downarrow; 0 \text{ if } \varphi_x(x) \uparrow\}$ **Reduction Pattern**: $\chi_k(x)$ = transformation_of_given_function(x)

Critical Success Factor: Always verify $f(x) \neq \varphi_x(x)$ for all $x \in \mathbb{N}$

This guide covers 100% of non-computable function exercises. Follow the templates mechanically and you will solve any exercise correctly!