1. შემდეგი რიცხვები წარმოადგინეთ პოზიციურ სისტემებში ფუძით 16,2,8 და10: 164, (65129123564211)8, (154)7.

164 = 101001002 = 2448 = A416 გადაყვანის მექანიზმი : ყველაზე დიდი ხარისხი მოთავსდა მე-8 ხარისხი, დარჩა 36 რომელიც დაიყო 32 და 4 ჩავწერე შესაბამისად ორობითში. ორობითი დავყავი 3 ნაწილად რომ გადამეყვანა 8 ობითში ხოლო 4 ად რომ გადამეყვანა 16 ობითში.

(65129123564211)8 = არ არის 8 ობითი სისტემის რიცხვი უნდა იყოს [0; 7] შუალედში

(154)7 = 1\*72 + 5\*71 + 4 = 8810 = 10110002 = 1308 = 5816

იგივე პრინციპით.

1. დაადგინეთ მთელი ნიშნიანი რიცხვის ათობითი მნიშვნელობა, თუ ამ რიცხვის ორობით წარმოდგენას შემდეგი სახე აქვს 1...10110011.

1...10110011

A+A- = -1

A = -(A- + 1)

A- = 0...01001100

A- + 1 = 0...01001101 = 26 + 23 + 22 + 1 = 64 + 8 +4 + 1 = 77

A = -77

1. იპოვეთ უსგ(45, 60 ) (ევკლიდეს გაფართოებული ალგორითმი).

60 = 1 \* 45 + 15

45 = 3 \* 15 + 0

უსგ(45; 60) = 15

1. ამოხსენით დიოფანტეს განტოლება 11x+43y=1 (კერძო ამონახსნის საპოვნელად გამოიყენეთ ევკლიდეს ალგორითმი).

(11, 43) = 1

43 = 11 \* 3 + 10

11 = 10 \* 1 + 1

10 = 1 \* 10 + 0

1 = 11 – 1 \* 10

1 = 11 – 1 \* (43 – 11 \* 3)

1 = 11 – 1 \* 43 + 11 \* 3

1 = 11 \* 4 + 43 \* (- 1)

კერძ ამონახსნი : x = 4; y = -1

1. წყლის ორი ჭურჭლიდან რომელთა მოცულბებია 4 და 21 ლიტრა, უნდა მივიღოთ ერთი ლიტრა წყალი. ჩვენ შეგვიძლია ერთი ჭურჭლიდან მეორე ჭურჭელში გადავასხათ წყალი ან საჭიროების შემთხვევაში დავცალოთ ეს ჭურჭელი (შეადგინეთ წრფივი ორიცნობიანი დიოფანტეს განტოლება და იპოვეთ მისი კერძო ამონახსნი).

4x + 21y = 1

(4, 21) = 1

21 = 4 \* 5 + 1

4 = 1 \* 4 + 0

21 – 4 \* 5 = 1

21 \* 1 + 4 \* (- 5) = 1

X = -5 y = 1

ცხადია ამ ამოხსნიდან პირდაპირ ჩანს რომ 21 ლიტრი ერთხელ რომ გავავსებთ. და შემდეგ 5 ჯერ გადავასხავთ 4 ლიტრიანში და 4 ;იტრიანს ყოველ ჯერზე დავცლით ბოლო გადასხვამზე 21 ლიტრიან ჭრჭელში 1 ლიტრი წყალი დაგვრჩება.

ცხადია ეს რიცხვები ურთერთ მარტივია მარა თვალსაჩნოებისათვის დავწწერე ორივეგან.

1. ჰანოეს კოშკი (ჩაწერა მხოლოდ რგოლების გამოყენებით).

ჰანოეს კოშკი არის ამოცანა რომელიც გულისხმობს ერთ ღეროზე ზომის მიხედვით დალაგებული რგოლების გადატანა მესამე ღეროზე მეორეს გამოყენებით ისე რომ პატარა რგოლს არ დაედოს დიდი რგოლი.

|  |  |  |
| --- | --- | --- |
| ღერო1 | ღერო 2 | ღერო 3 |
| abc | - | - |
| bc | - | a |
| c | b | a |
| c | ab | - |
| - | ab | c |
| a | b | c |
| a | - | bc |
| - | - | abc |

სადემონსტრაციოდ ასე იქნება 3 რგოლის შემთხვევაში.

ზოგად პრინციპს რაც შეეხება. გასათვალისწინებელია რომ შეგვიძლია დავიჭიროთ რეკურსიული ლოგიკა გადატანებში.

თუ გვინდა n რაოდენობის რგოლები გადავიტანოთ ჯერ n – 1 ცალი რგოლი უნდა გადავიტანოთ დამხმარე რგოლზე, შემდგომ ყველაზე დიდი რგოლი გადავიტანოთ დანიშნულების ადგილას და n – 1 რგოლი დავადგათ თავზე.

n – 1 ის შემთხვევაშიც იგივე ამოცანის წინაშე ვართ, რონ ჯერ n – 2 ცალი უნდა გადავიტანოთ შემდგომ ბოლო და უკან დავადგათ დანარჩენები. შესაბამისად ეს არის ჰანოეს კოშკის რეკურსიულობა. ხოლო რაც რეალურად ვიცით ეს არის ერთი დისკის გადატანის ალგორითმი. კერძოდ როდესაც რეკურსიული ალგორითმი მიაღწევს ერთ ცალ დისკს ჩვენ უბრალოდ მისი გადატანა უნდა მოვახდინოთ სადაც გვინდა. ამიტომაც ერთი დისკის გადატანა იქნება ჩვენი რეკურსიის გაჩერების პირობა და როგორც კი ეს ამოიხსნება ჯაჭვურად წინა წევრებიც ამოიხსნება.

განზოგადებულად კი შეგვიძ₾ია დავწეროთ შემდეგ ნაირად რეკურსიული ფუნქციისათვის

(n – 1, საწყისი, დამხმარე , მიზანი)

(n, საწყისი, მიზანი)

(n – 1, დამხმარე, მიზანი, საწყისი)

შესაბამისად ზემოთ როგოც ავღწერე. ჯერ n – 1 გადაგვაქვს. შემდენ ბოლო

დაბრუნებუს პირობა კიდევ იქნება როდესაც n = 1;

1. შეადგინეთ პროგრამა, რომელშიც გათვალისწინებულია ერთი მთელი რიცხვის შეტანა და დაადგენს არის თუ არა ეს რიცხვი პოლინდრომი (მაგ. 123 არ არის პოლინრომი, 121 არის პოლინდრომი).

შეადგინეთ პროგრამა, რომელშიც გათვალისწინებულია 2 მთელი რიცხვის,  N და M  (1<=N<=1018) შეტანა, თუ ეს შეტანილი რიცხვებს აქვს განსხვავებული ლუწ-კენტოვნების მნიშვნელობა დაბეჭდეთ 1, წინააღმდეგ შემთხვევაში დაბეჭდეთ 0. (ამოცანა უნდა ამოიხსნას ბიტური ოპერაციების გამოყენებით).

იხილეთ cpp ფაილი.