



جواب سوال های دستور کار ۲

شبه کد ها و الگوریتم

۱۴) الگوریتم اقلیدسی بر مبنای تقسیم به صورت تابع:

The image shows a Scratch code editor with a project titled "findGCD". The code is written in JavaScript and implements the Euclidean algorithm for finding the Greatest Common Divisor (GCD) of two numbers, a and b.

Code Structure:

- Function Definition:** A function block "to findGCD with: a, b" is defined. It includes:
 - A "set var to 0" block.
 - A "repeat while" loop with the condition "b > 0".
 - Inside the loop, a "do" block contains:
 - "set var to a"
 - "set a to b"
 - An "if" block with the condition "remainder of var ÷ b = 0".
 - Inside the "if" block, a "print b" block is present.
 - "set b to remainder of var ÷ b"
 - A "return" block at the end of the function.
- Function Usage:** Below the function definition, the following code is executed:
 - "set a to 15"
 - "set b to 10"
 - A "print" block with the text "findGCD with:" followed by two input fields for "a" and "b".

The code is written in a Scratch-style block-based language, with blocks color-coded by category: Logic (purple), Loops (green), Math (blue), Text (green), Lists (purple), Colour (brown), Variables (purple), and Functions (purple).

الگوریتم اقلیدسی بر مبنای تقسیم به صورت تابع خروجی:

The image shows a Scratch code editor with a JavaScript tab selected. The code defines a function `findGCD` that takes two arguments, `a` and `b`. The function uses a `repeat while` loop to calculate the GCD using the Euclidean algorithm. It sets a variable `var` to 0, then enters a loop that continues as long as `b` is greater than 0. Inside the loop, it sets `var` to `a`, sets `a` to `b`, and checks if the remainder of `a` divided by `b` is 0. If so, it prints `b` and sets `b` to the remainder of `a` divided by `b`. After the loop, it returns `var`. Below the function, it sets `a` to 15, sets `b` to 10, and prints the result of `findGCD` with arguments `a` and `b`.

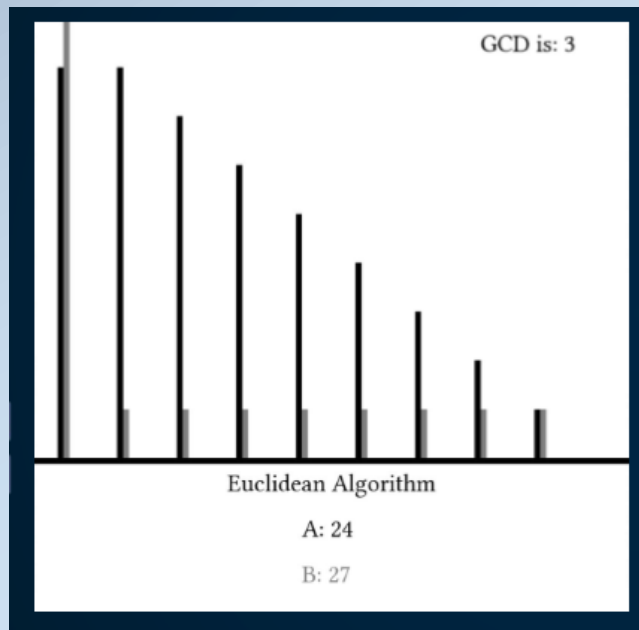
A dialog box titled "This site says..." is displayed in the center of the screen, showing the number 5 and an OK button.

```
function findGCD(a, b) {
  var = 0;
  repeat while (b > 0) {
    do {
      set var to a;
      set a to b;
      if (remainder of a by b = 0) {
        do {
          print b;
          set b to remainder of a by b;
        }
      }
    }
  }
  return var;
}

set a to 15;
set b to 10;
print findGCD(a, b);
```

۱۵) مراحل اجرای دو الگوریتم اقلیدسی معرفی شده برای دو عدد ۲۴ و ۲۷ و مقایسه ی نمودار ها:

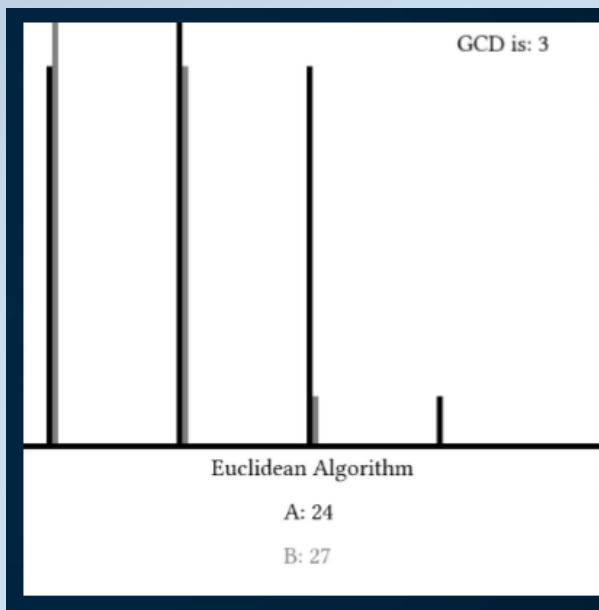
- الگوریتم اقلیدسی بر پایه ی تفریق: مشاهده می شود که اگر مقدار $b=27$ و مقدار $a=24$ فرض کنیم، بعد از ۱ بار اجرای حلقه ی `while`، مقدار b بسیار کم تر از a میشود و برای اینکه $a=b$ شود تا الگوریتم خاتمه پیدا کند، باید در کل ۸ مرتبه بدنه ی حلقه ی `while` اجرا شود و در هر مرحله مقدار b ثابت مانده است و مقدار a تغییر می کند. در نمودار میله خاکستری مقدار b و میله ی مشکی مقدار a را نشان میدهد.



a	b	
۲۴	۲۷	← مقادیر ابتدایی
۲۴	۳	← $b = b - a$
۲۱	۳	← $a = a - b$
۱۸	۳	← $a = a - b$
۱۵	۳	← $a = a - b$
۱۲	۳	← $a = a - b$
۹	۳	← $a = a - b$
۶	۳	← $a = a - b$
۳	۳	← $a = a - b$

مراحل اجرای دو الگوریتم اقلیدسی معرفی شده برای دو عدد ۲۴ و ۲۷ و مقایسه ی نمودار ها:

- **الگوریتم اقلیدسی بر پایه ی تقسیم:** مشاهده می شود که اگر مقدار $b=27$ و مقدار $a=24$ فرض کنیم، بعد از ۱ بار اجرای حلقه ی `while`، مقادیر a و b با یک دیگر جابجا می شود و در دور بعد اجرای حلقه باقی مانده ی تقسیم `var` به b در b قرار میگیرد که مقدار ۳ دارد و مقدار بعدی `var` بر ۳ بخش پذیر است پس b ب.م.م است و مقدارش بعد از چاپ شدن • میشود. پس در کل ۳ مرتبه بدنه ی حلقه ی `while` اجرا شده است. در نمودار میله خاکستری مقدار b و میله ی مشکی مقدار a را نشان میدهد.
- پس با مقایسه ی این الگوریتم با الگوریتم بر پایه ی تفریق می بینیم با اینکه الگوریتم اقلیدسی بر پایه ی تفریق ساده تر است ولی تعداد مراحل این الگوریتم کم تر است (به خصوص برای اعداد بزرگ) و زود تر به جواب می رسد پس عملکرد بهتری دارد و بهینه تر است.



a	b	var
24	27	0
27	24	24
24	3	27
3	0	24

مقادیر ابتدای

$\text{rem}(24, 3) = 0$

Print b

١٨ ساعة

Blocks

JavaScriptPythonPHPLuaDartXML

Logic

Loops

Math

Text

Lists

Colour

Variables

Functions

set h to 0

set m to 0

set s to 0

repeat while h < 24

do

repeat while m < 60

do

repeat while s < 60

do

set item to 0

to item append text h

to item append text “ : ”

to item append text m

to item append text “ : ”

to item append text s

set s to s + 1

set m to m + 1

set s to 0

set h to h + 1

set m to 0

🗑️

↺

▶️

🎯

+

-

🗑️

٢٠) سوال مبنا

Let $a = 45$

Let $b = 2$

Repeat

45 divided by b is $45/2 = 22$ remainder 1. So $d=22$ and $r=1$. So $m= 1$ and the new a is 22.

22 divided by b is $22/2 = 11$ remainder 0. So $d=11$ and $r=1$. So $m= 01$ and the new a is 11.

11 divided by b is $11/2 = 5$ remainder 1. So $d=5$ and $r=1$. So $m= 101$ and the new a is 5.

5 divided by b is $5/2 = 2$ remainder 1. So $d=2$ and $r=1$. So $m= 1101$ and the new a is 2.

2 divided by b is $2/2 = 1$ remainder 0. So $d=1$ and $r=0$. So $m= 01101$ and the new a is 1.

1 divided by b is $1/2 = 0$ remainder 1. So $d=0$ and $r=1$. So $m=101101$ and the new a is 0. So $45_{10} = 101101_2$

- Logic
- Loops
- Math
- Text
- Lists
- Colour
- Variables
- Functions

```

set a to 45
set b to 2
set m to create empty list
set index to 1
repeat while a > 0
do
  in list m insert at # index as remainder of a ÷ b
  set index to index + 1
  set a to round down a ÷ b
set item to 0
repeat while index > 0
do
  to item append text in list m get # index
  set index to index - 1
print item

```

Run the program defined by the blocks in the

خروجی

code/index.html

blockly-demo.appspot.com says

0undefined101101

OK

۲۲ سوال عددبازی

- قسمت های اضافه شده به شبه کد اسلاید ۲۷

Blocks

JavaScript

Python

PHP

Lua

Dart

XML



Logic

Loops

Math

Text

Lists

Colour

Variables

Functions

