4.13.83

AI25BTECH11014 - Gooty Suhas

October 1, 2025

Question

Let a, b, c be distinct non-negative numbers. If the vectors

$$\mathbf{A} = \begin{pmatrix} a \\ a \\ c \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} a \\ c \\ b \end{pmatrix}$$

lie in a plane, then c is:

Options:

a) Arithmetic Mean of a and b

b) Geometric Mean of a and b

c) Harmonic Mean of a and b

d) Equal to zero

Difference Vectors

$$\mathbf{A}-\mathbf{B}=\begin{pmatrix} a-1 \\ a \\ c-1 \end{pmatrix}, \quad \mathbf{C}-\mathbf{B}=\begin{pmatrix} a-1 \\ c \\ b-1 \end{pmatrix}$$

Initial matrix:

$$\mathbf{M}=\begin{pmatrix} a-1 & a-1 \\ a & c \\ c-1 & b-1 \end{pmatrix}$$

Row Operation 1

Apply $R_2 \leftarrow R_2 - R_1$:

$$R_2 = \begin{pmatrix} 1 & c-a \end{pmatrix}$$

Now $\mathbf{M} =$

$$\begin{pmatrix} a-1 & a-1 \\ 1 & c-a \\ c-1 & b-1 \end{pmatrix}$$

Row Operation 2

Apply $R_3 \leftarrow R_3 - R_1$:

$$R_3 = \begin{pmatrix} c - a & b - a \end{pmatrix}$$

Now $\mathbf{M} =$

$$\begin{pmatrix} a - 1 & a - 1 \\ 1 & c - a \\ c - a & b - a \end{pmatrix}$$

Row Operation 3

Eliminate R_3 using R_2:

$$R_3 \leftarrow R_3 - (c-a) \cdot R_2 \Rightarrow \begin{pmatrix} 0 & b-a-(c-a)^2 \end{pmatrix}$$

Now $\mathbf{M} =$

$$\begin{pmatrix} a-1 & a-1 \\ 1 & c-a \\ 0 & b-a-(c-a)^2 \end{pmatrix}$$

Collinearity Condition

For collinearity:

$$b - a - (c - a)^2 = 0 \Rightarrow (c - a)^2 = b - a \Rightarrow c = a + \sqrt{b - a}$$

Try $c = \sqrt{ab}$:

$(c - a)^2 = ab - 2a\sqrt{ab} + a^2 \Rightarrow$ Set equal to $b - a \Rightarrow ab - 2a\sqrt{ab} + a^2 = b - a$

Final Answer

$$\boxed{c = \text{Geometric Mean of } a \text{ and } b} \quad \Rightarrow \quad \boxed{\text{Option (b)}}$$

Python Code — SymPy (1/2)

```
from sympy import Matrix, symbols

a, b, c = symbols('a␣b␣c')
A = Matrix([a, a, c])
B = Matrix([1, 0, 1])
C = Matrix([a, c, b])

AB = A - B
CB = C - B

M = Matrix.hstack(AB, CB)
```

Python Code — SymPy (2/2)

```
rref, _ = M.rref()

print("Row␣Echelon␣Form:")
print(rref)

# Check condition
diff = rref[2,1]
if diff == 0:
    print("Vectors␣lie␣in␣a␣plane")
else:
    print("Not␣coplanar")
```

C Code (1/2)

```
#include <stdio.h>
#include <math.h>

int main() {
    double a, b;
    scanf("%lf␣%lf", &a, &b);

    double c = sqrt(a * b);

    double AB[3] = {a - 1, a, c - 1};
    double CB[3] = {a - 1, c, b - 1};
```

C Code (2/2)

```
    double R2_0 = AB[1] - AB[0];
    double R2_1 = CB[1] - CB[0];

    double R3_0 = AB[2] - AB[0];
    double R3_1 = CB[2] - CB[0];

    double final = R3_1 - (R3_0 * R2_1);

    if (final == 0)
        printf("Vectors␣lie␣in␣a␣plane\n");
    else
        printf("Not␣coplanar\n");

    return 0;
}
```

Python Code — Executable Runner (1/2)

```
import subprocess

# Input values for a and b
a = 4
b = 9

# Prepare input string
input_str = f"{a}␣{b}\n"

# Run compiled C binary
result = subprocess.run(
    ['./coplanar_check'], # executable name
    input=input_str,
```

Python Code — Executable Runner (2/2)

```
    capture_output=True,
    text=True
)

# Output result
output = result.stdout.strip()
print("Result␣from␣C␣program:")
print(output)

# Optional: check return code
if result.returncode != 0:
    print("Execution␣failed")
```