# FIND\_BITS

שם: רונן אברהם וול

ת"ז: 215757477

הסבר: קלטתי שני unsighned long מהמשתמש לאחר מכן השתמשתי בפעולה ביטוויז של AND (& כפל לוגי) וכך נשארו רק הביטים שהיו דולקים בשני המשתנים.

אחרי זה קראתי לפונקציה רקורסיבית שמחשבת את כמות הביטים הדולקים ומדפיסה כל ביט בסדר הפוך ללא ירידת שורה כך שבסוף מוצגים הביטים המשותפים כ 1 והכבויים כ0.

קוד:

    #include <stdio.h>

    unsigned long countBits(unsigned long commonBits)

    {

        int returnValue = 0;

        if(commonBits != 0)

        {

            returnValue = countBits(commonBits / 2);

            if(commonBits % 2)

            {

                printf("1");

                returnValue++;

            }

            else

            {

                printf("0");

            }

        }

        return returnValue;

    }

    int main()

    {

        /\*\*

         \* This program counts how many common 1 bits two unsigned long integers have.

         \*

         \* var x: The first unsigned long integer input by the user.

         \* var y: The second unsigned long integer input by the user.

         \*

         \* type x: unsigned long

         \* type y: unsigned long

         \*

         \* return: Prints the number of common 1 bits between x and y.

         \* rtype: int

         \*/

        unsigned long x;

        unsigned long y;

        printf("Enter an unsigened long x and y to count how many common 1 bits they have:\n");

        scanf("%lu %lu", &x, &y);

        unsigned long commonBits = x & y;

        printf("the common bits of x and y:\t");

        if (!commonBits) // ensure that there is still a print even if there are no common on bits

            printf("0");

        unsigned long count = countBits(commonBits);

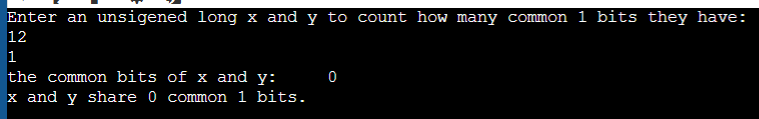
        printf("\n");

        printf("x and y share %lu common 1 bits.\n", count);

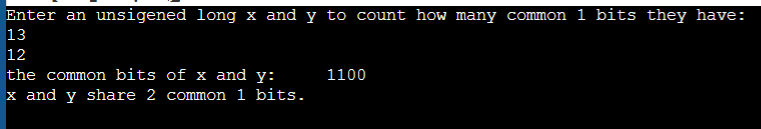
        return 0;

    }

פלט 1:



פלט 2:



פלט 3:

תמונה שמכילה טקסט, תוכנה, תכונות מולטימדיה, גופן

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.