recursive function algebra 2

recursive function algebra 2 is a pivotal concept that integrates algebraic principles with the foundational understanding of functions. In Algebra 2, students encounter recursive functions as a means to define sequences and series, where each term is derived from its predecessors. This article delves into the essence of recursive functions, their applications, and how they relate to the broader curriculum of Algebra 2. We will explore various examples, types of recursive functions, and techniques for solving them, providing a comprehensive guide for students and educators alike.

The following sections will outline the core components of recursive functions, their mathematical significance, and methods to analyze and implement them effectively in problem-solving scenarios.

Understanding Recursive Functions

Recursive functions are defined by a relation that relates each term in a sequence to one or more previous terms. This definition allows for the construction of sequences where the nth term can be expressed as a function of preceding terms. In Algebra 2, students encounter recursive functions in various forms, particularly in sequences and series.

Definition and Characteristics

A recursive function can be described using an initial condition and a recursive rule. The initial condition provides the first term of the sequence, while the recursive rule defines how subsequent terms are generated. For example, the Fibonacci sequence is a classic example of a recursive function:

- **Initial Condition:** F(0) = 0, F(1) = 1

- **Recursive Rule:** $F(n) = F(n-1) + F(n-2)$ for $n \geq 2$

This means that each term in the Fibonacci sequence is the sum of the two preceding terms, starting from 0 and 1.

Importance in Algebra 2

Understanding recursive functions is crucial for students in Algebra 2 as they provide a bridge between algebraic expressions and sequences. They help students grasp concepts of limits, convergence, and mathematical modeling. Recursive functions also lay the groundwork for more complex topics in calculus and computer science.

Types of Recursive Functions

There are various types of recursive functions that students will encounter in Algebra 2, including linear recursion and nonlinear recursion. Each type has its own characteristics and uses.

Linear Recursive Functions

Linear recursive functions are those where each term is a linear combination of previous terms. These functions can be expressed in the form:

$T(n) = a\,T(n-1) + b$, where a and b are constants.

For example, the sequence defined by $T(n) = 2\,T(n-1) + 3$ with $T(0) = 1$ is a linear recursive function.

Nonlinear Recursive Functions

Nonlinear recursive functions involve more complex relationships between terms. An example is the sequence defined by $T(n) = T(n-1)^2 + 1$. Nonlinear functions can exhibit exponential growth and are often more challenging to solve.

How to Solve Recursive Functions

Solving recursive functions often involves finding a closed-form expression that allows for direct computation of any term without needing to compute all preceding terms. Several strategies can be employed to achieve this.

Finding Closed-Form Solutions

To find a closed-form solution, one can use techniques such as:

- **Substitution:** Substitute previous terms into the recursive relation to derive a pattern.
- **Iteration:** Expand the recursive formula step-by-step to identify a general formula.
- **Mathematical Induction:** Prove that a proposed closed-form solution holds for all integers.

Example of Solving a Recursive Function

Consider the recursive function defined as follows:

T(n) = 3 T(n-1) - 2 with T(0) = 5. To solve this:

1. Identify the pattern by calculating the first few terms:
2. T(1) = 3 5 - 2 = 13
3. T(2) = 3 13 - 2 = 37
4. T(3) = 3 37 - 2 = 109

After observing the pattern, one might conjecture a closed form and prove it using induction.

Applications of Recursive Functions in Algebra 2

Recursive functions have a variety of applications within Algebra 2. They are used to model real-world situations, analyze sequences, and even in computer algorithms.

Modeling Real-World Scenarios

Students can use recursive functions to model phenomena such as population growth, financial calculations, and more. For instance, if a population of rabbits doubles every year, the growth can be modeled recursively. If P(n) is the population in year n, then:

- P(0) = initial population
- P(n) = 2 P(n-1) for n ≥ 1

Analyzing Sequences and Series

Many sequences encountered in Algebra 2 can be expressed recursively. Understanding these functions allows students to derive formulas for series sums, find convergence, and explore infinite series.

Common Mistakes and Misunderstandings

Students often face challenges when learning about recursive functions. Recognizing common mistakes can help prevent confusion.

Misunderstanding Initial Conditions

One frequent mistake is neglecting to correctly apply the initial condition. The initial term is critical for the proper generation of subsequent terms in a recursive sequence.

Overlooking the Recursive Rule

Another common issue is misunderstanding the recursive rule. Students must ensure they apply the rule consistently to derive correct terms.

Conclusion

Recursive function algebra 2 is an essential topic that plays a significant role in understanding sequences and their applications. By grasping the concepts of recursion, students can enhance their problem-solving skills and apply these functions in various mathematical and real-world contexts. Mastering recursive functions paves the way for more advanced studies in mathematics, including calculus and computer science, making it a critical component of the Algebra 2 curriculum.

Q: What is a recursive function?

A: A recursive function is a function that is defined in terms of itself, allowing for the computation of terms in a sequence based on previous terms. It typically includes an initial condition and a recursive rule.

Q: How do I identify a recursive function?

A: To identify a recursive function, look for a relation that defines each term based on one or more preceding terms along with an initial condition that specifies the first term of the sequence.

Q: Can all sequences be expressed recursively?

A: Most sequences can be expressed recursively, though some may be more straightforwardly defined using a closed-form expression. Recursive definitions highlight the relationships between terms.

Q: What are some common applications of recursive functions?

A: Recursive functions are used in various fields, such as computer science for algorithms, in modeling population growth, and in financial calculations for compound interest.

Q: How do you solve a recursive function?

A: Solving a recursive function may involve finding a closed-form expression through techniques like substitution, iteration, or mathematical induction to derive a direct formula for any term.

Q: What is the Fibonacci sequence in the context of recursion?

A: The Fibonacci sequence is a classic example of a recursive function where each term is the sum of the two preceding terms, defined as $F(n) = F(n-1) + F(n-2)$ with initial conditions $F(0) = 0$ and $F(1) = 1$.

Q: What are linear and nonlinear recursive functions?

A: Linear recursive functions relate terms through linear combinations, while nonlinear recursive functions involve more complex relationships that can lead to exponential growth.

Q: What mistakes should I avoid when studying recursive functions?

A: Common mistakes include neglecting the initial condition and misunderstanding the recursive rule, which are crucial for correctly generating terms in a sequence.

Q: How do recursive functions relate to sequences and series?

A: Recursive functions define sequences, where each term is generated based on previous terms. Understanding these functions is key to analyzing and summing series in algebra.

Q: Why are recursive functions important in mathematics?

A: Recursive functions are important because they provide insight into sequences, allow for the modeling of real-world phenomena, and serve as a foundation for more advanced mathematical concepts, including calculus.

Recursive Function Algebra 2

recursive function algebra 2: *Recursively Enumerable Sets and Degrees* Robert I. Soare, 1999-11-01 ...The book, written by one of the main researchers on the field, gives a complete account of the theory of r.e. degrees. .... The definitions, results and proofs are always clearly motivated and explained before the formal presentation; the proofs are described with remarkable clarity and conciseness. The book is highly recommended to everyone interested in logic. It also provides a useful background to computer scientists, in particular to theoretical computer scientists. Acta Scientiarum Mathematicarum, Ungarn 1988 ...The main purpose of this book is to introduce the reader to the main results and to the intricacies of the current theory for the recurseively enumerable sets and degrees. The author has managed to give a coherent exposition of a rather complex and messy area of logic, and with this book degree-theory is far more accessible to students and logicians in other fields than it used to be. Zentralblatt für Mathematik, 623.1988

recursive function algebra 2: Classical Recursion Theory P. Odifreddi, 1992-02-04 1988 marked the first centenary of Recursion Theory, since Dedekind's 1888 paper on the nature of number. Now available in paperback, this book is both a comprehensive reference for the subject and a textbook starting from first principles. Among the subjects covered are: various equivalent approaches to effective computability and their relations with computers and programming languages; a discussion of Church's thesis; a modern solution to Post's problem; global properties of Turing degrees; and a complete algebraic characterization of many-one degrees. Included are a number of applications to logic (in particular Gödel's theorems) and to computer science, for which Recursion Theory provides the theoretical foundation.

recursive function algebra 2: Recursion Theory and Complexity Marat M. Arslanov, Steffen Lempp, 2014-10-10 The series is devoted to the publication of high-level monographs on all areas of mathematical logic and its applications. It is addressed to advanced students and research mathematicians, and may also serve as a guide for lectures and for seminars at the graduate level.

recursive function algebra 2: Algebraic Methods in Semantics M. Nivat, John C. Reynolds, 1985 This book, which contains contributions from leading researchers in France, USA and Great Britain, gives detailed accounts of a variety of methods for describing the semantics of programming languages, i.e. for attaching to programs mathematical objects that encompass their meaning. Consideration is given to both denotational semantics, where the meaning of a program is regarded as a function from inputs to outputs, and operational semantics, where the meaning includes the sequence of states or terms generated internally during the computation. The major problems considered include equivalence relations between operational and denotational semantics, rules for obtaining optimal computations (especially for nondeterministic programs), equivalence of programs, meaning-preserving transformations of programs and program proving by assertions. Such problems are discussed for a variety of programming languages and formalisms, and a wealth of mathematical tools is described.

recursive function algebra 2: Formal Systems and Recursive Functions Lev D. Beklemishev, 2000-04-01 Formal Systems and Recursive Functions

recursive function algebra 2: Matemati?eskaja Logika, Teorija Algoritmov i Teorija Mno�estv S. I. Adi�a�n, 1977 Papers celebrating Petr Sergeevič Novikov and his work in descriptive set theory and algorithmic problems of algebra.

recursive function algebra 2: *Recursive Function Theory and Logic* Ann Yasuhara, 1971

recursive function algebra 2: Recursion Theory Anil Nerode, Richard A. Shore, 1985

recursive function algebra 2: Computability In Context: Computation And Logic In The Real World S Barry Cooper, Andrea Sorbi, 2011-02-25 Computability has played a crucial role in mathematics and computer science, leading to the discovery, understanding and classification of decidable/undecidable problems, paving the way for the modern computer era, and affecting deeply our view of the world. Recent new paradigms of computation, based on biological and physical models, address in a radically new way questions of efficiency and challenge assumptions about the so-called Turing barrier.This volume addresses various aspects of the ways computability and theoretical computer science enable scientists and philosophers to deal with mathematical and real-world issues, covering problems related to logic, mathematics, physical processes, real computation and learning theory. At the same time it will focus on different ways in which computability emerges from the real world, and how this affects our way of thinking about everyday computational issues./a

recursive function algebra 2: Mathematical Foundations of Computer Science 1981 J. Gruska, M. Chytil, 1981-08

recursive function algebra 2: Handbook of Computability and Complexity in Analysis Vasco Brattka, Peter Hertling, 2021-06-04 Computable analysis is the modern theory of computability and complexity in analysis that arose out of Turing's seminal work in the 1930s. This was motivated by questions such as: which real numbers and real number functions are computable, and which mathematical tasks in analysis can be solved by algorithmic means? Nowadays this theory has many different facets that embrace topics from computability theory, algorithmic randomness, computational complexity, dynamical systems, fractals, and analog computers, up to logic, descriptive set theory, constructivism, and reverse mathematics. In recent decades computable analysis has invaded many branches of analysis, and researchers have studied computability and complexity questions arising from real and complex analysis, functional analysis, and the theory of differential equations, up to (geometric) measure theory and topology. This handbook represents the first coherent cross-section through most active research topics on the more theoretical side of the field. It contains 11 chapters grouped into parts on computability in analysis; complexity, dynamics, and randomness; and constructivity, logic, and descriptive complexity. All chapters are written by leading experts working at the cutting edge of the respective topic. Researchers and graduate students in the areas of theoretical computer science and mathematical logic will find systematic introductions into many branches of computable analysis, and a wealth of information and references that will help them to navigate the modern research literature in this field.

recursive function algebra 2: *Interpretations of Probability* Andrei Khrennikov, 2009-02-26 This is the first fundamental book devoted to non-Kolmogorov probability models. It provides a mathematical theory of negative probabilities, with numerous applications to quantum physics, information theory, complexity, biology and psychology. The book also presents an interesting model of cognitive information reality with flows of information probabilities, describing the process of thinking, social, and psychological phenomena.

recursive function algebra 2: Typed Lambda Calculi and Applications Pawel Urzyczyn, 2005-04-07 This book constitutes the refereed proceedings of the 7th International Conference on Typed Lambda Calculi and Applications, TLCA 2005, held in Nara, Japan in April 2005. The 27 revised full papers presented together with 2 invited papers were carefully reviewed and selected from 61 submissions. The volume reports research results on all current aspects of typed lambda calculi, ranging from theoretical and methodological issues to applications in various contexts.

recursive function algebra 2: PROGRESS IN PHYSICS, Vol. 15. The Journal on Advanced Studies in Theoretical and Experimental Physics, including Related Themes from Mathematics Dmitri Rabounski, Florentin Smarandache, Larissa Borissova, Progress in Physics has been created for rapid publications on advanced studies in theoretical and experimental physics, including related themes from mathematics and astronomy.

recursive function algebra 2: *Machines, Computations, and Universality* Jérôme Durand-Lose, Maurice Margenstern, 2007-08-23 This book constitutes the refereed proceedings of the 5th International Conference on Machines, Computations, and Universality, MCU 2007, held in Orleans, France, September 2007. The 18 revised full papers presented together with nine invited papers cover Turing machines, register machines, word processing, cellular automata, tiling of the plane, neural networks, molecular computations, BSS machines, infinite cellular automata, real machines, and quantum computing.

recursive function algebra 2: Countable Boolean Algebras and Decidability Sergey Goncharov, 1997-01-31 This book describes the latest Russian research covering the structure and algorithmic properties of Boolean algebras from the algebraic and model-theoretic points of view. A significantly revised version of the author's Countable Boolean Algebras (Nauka, Novosibirsk, 1989), the text presents new results as well as a selection of open questions on Boolean algebras. Other current features include discussions of the Kottonen algebras in enrichments by ideals and automorphisms, and the properties of the automorphism groups.

recursive function algebra 2: Computability and Models Barry S. Cooper, 2012-12-06 Science involves descriptions of the world we live in. It also depends on nature exhibiting what we can best describe as a high aLgorithmic content. The theme running through this collection of papers is that of the interaction between descriptions, in the form of formal theories, and the algorithmic content of what is described, namely of the modeLs of those theories. This appears most explicitly here in a number of valuable, and substantial, contributions to what has until recently been known as 'recursive model theory' - an area in which researchers from the former Soviet Union (in particular Novosibirsk) have been pre-eminent. There are also articles concerned with the computability of aspects of familiar mathematical structures, and - a return to the sort of basic underlying questions considered by Alan Turing in the early days of the subject - an article giving a new perspective on computability in the real world. And, of course, there are also articles concerned with the classical theory of computability, including the first widely available survey of work on quasi-reducibility. The contributors, all internationally recognised experts in their fields, have been associated with the three-year INTAS-RFBR Research Project Com putability and Models (Project No. 972-139), and most have participated in one or more of the various international workshops (in Novosibirsk, Heidelberg and Almaty) and otherresearch activities of the network.

recursive function algebra 2: Construction and Analysis of Cryptographic Functions Lilya Budaghyan, 2015-01-06 This book covers novel research on construction and analysis of optimal cryptographic functions such as almost perfect nonlinear (APN), almost bent (AB), planar and bent functions. These functions have optimal resistance to linear and/or differential attacks, which are the two most powerful attacks on symmetric cryptosystems. Besides cryptographic applications, these functions are significant in many branches of mathematics and information theory including coding theory, combinatorics, commutative algebra, finite geometry, sequence design and quantum information theory. The author analyzes equivalence relations for these functions and develops several new methods for construction of their infinite families. In addition, the book offers solutions to two longstanding open problems, including the problem on characterization of APN and AB functions via Boolean, and the problem on the relation between two classes of bent functions.

recursive function algebra 2: I Love Ruby Karthikeyan A K, Starting out Ruby programming, then probaly this book is the most easiest way to learn it. This book comes complete with tens of example code.

recursive function algebra 2: Dictionary of Logic as Applied in the Study of Language W. Marciszewski, 2013-06-29 1. STRUCTURE AND REFERENCES 1.1. The main part of the dictionary consists of alphabetically arranged articles concerned with basic logical theories and some other selected topics. Within each article a set of concepts is defined in their mutual relations. This way of defining concepts in the context of a theory provides better understand ing of ideas than that provided by isolated short defmitions. A disadvantage of this method is that it takes more time to look something up inside an extensive article. To reduce this disadvantage the following measures have been adopted. Each article is divided into numbered sections, the numbers, in boldface type, being addresses to which we refer. Those sections of larger articles which are divided at the first level, i.e. numbered with single numerals, have titles. Main sections are further subdivided, the subsections being numbered by numerals added to the main section number, e.g. I, 1.1, 1.2, ... , 1.1.1, 1.1.2, and so on. A comprehensive subject index is supplied together with a glossary. The aim of the latter is to provide, if possible, short defmitions which sometimes may prove sufficient. As to the use of the glossary, see the comment preceding it.

Related to recursive function algebra 2

RECURSIVE□□ (□□)□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□;□□□□□□□□□

recursive - □□ □□ 7. The recursive nature of the approach allows decomposition to occur as many times as needed to identify atomic system elements. □ □□ □□ □ □□ □□ □ □□ □□ □□ □ □□ □ □

□□ - □□□□□□□□□□□□ □ □□ □□□ □□□□□□ □□□□□□□□□□ □□□ □□

recursive□□□□□_recursive□□□_□□_□□_□□_□□_□□ □□□177□□ □□□□□□□□□□□□□□AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□

recursive□□□□□□_□□□□ recursive□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□"-r"□□□□□□□□□□□□

□□□□□Recursion□□□□□ - CSDN□□ □□□□□ □ C □□□□□□Recursion□□□□ □□□□ □□□ □□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□□□□□□□□□
RECURSIVE □□ | □□□□□□□ - Collins Online Dictionary RECURSIVE □□: reapplying the same formula or algorithm to a number or result in order to generate the | □□□□□□□□□□□
□□□□|□□-□□□□ recursive□□□□□_recursive□□□□□□□_recursive□□□_recursive □□□□□□□□□recursive□□□□□□□□□□□□□recursive□□□□□recursive□□□recursive□□□□recursive□□□□recursive□□□□
recursive□□□□□_recursive□□□□_□□_□□_□□_□□_□ □□□□,□□□□□□□□,□□□□recursive□□□□□,recursive□□□□□,recursive□□□□□,□□□□□□□□□□□
RECURSIVE□□□□□□□□□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□□□□□□1. involving doing or saying the same thing several times in order to produce a particular result□□□□□□
RECURSIVE□□ (□□)□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□;□□□□□□□□
recursive - □□ □□ 7. The recursive nature of the approach allows decomposition to occur as many times as needed to identify atomic system elements. □ □□ □□ □ □□ □□ □ □□ □□ □□ □ □□ □ □
□□ - □□□□□□□□□□□□ □ □□ □□□ □□□□□□ □□□□□□□□□□ □□□ □□
recursive□□□□□_recursive□□□_□□_□□_□□_□□_□□ □□□177□□□□□□□□□□□□□□□AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□
recursive□□□□□□_□□□□ recursive□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□"-r"□□□□□□□□□□□
□□□□□Recursion□□□□□ - CSDN□□ □□□□□ □ C □□□□□□Recursion□□□□ □□□□ □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□□□□□□□□□
RECURSIVE □□ | □□□□□□□ - Collins Online Dictionary RECURSIVE □□: reapplying the same formula or algorithm to a number or result in order to generate the | □□□□□□□□□□□
□□□□|□□-□□□□ recursive□□□□□_recursive□□□□□□□_recursive□□□_recursive □□□□□□□□□recursive□□□□□□□□□□□□□recursive□□□□□recursive□□□recursive□□□□recursive□□□□recursive□□□□
recursive□□□□□_recursive□□□□_□□_□□_□□_□□_□ □□□□,□□□□□□□□,□□□□recursive□□□□□,recursive□□□□□,recursive□□□□□,□□□□□□□□□□□
RECURSIVE□□□□□□□□□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□□□□□□1. involving doing or saying the same thing several times in order to produce a particular result□□□□□□
RECURSIVE□□ (□□)□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□;□□□□□□□□
recursive - □□ □□ 7. The recursive nature of the approach allows decomposition to occur as many times as needed to identify atomic system elements. □ □□ □□ □ □□ □□ □ □□ □□ □□ □ □□ □ □
□□ - □□□□□□□□□□□□ □ □□ □□□ □□□□□□ □□□□□□□□□□ □□□ □□
recursive□□□□□_recursive□□□_□□_□□_□□_□□_□□ □□□177□□□□□□□□□□□□□□□AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□
recursive□□□□□□_□□□□ recursive□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□"-r"□□□□□□□□□□□
□□□□□Recursion□□□□□ - CSDN□□ □□□□□ □ C □□□□□□Recursion□□□□ □□□□ □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□□□□□□□□□
RECURSIVE □□ | □□□□□□□ - Collins Online Dictionary RECURSIVE □□: reapplying the same formula or algorithm to a number or result in order to generate the | □□□□□□□□□□□
□□□□|□□-□□□□ recursive□□□□□_recursive□□□□□□□_recursive□□□_recursive □□□□□□□□□recursive□□□□□□□□□□□□□recursive□□□□□recursive□□□recursive□□□□recursive□□□□recursive□□□□
recursive□□□□□_recursive□□□□_□□_□□_□□_□□_□ □□□□,□□□□□□□□,□□□□recursive□□□□□,recursive□□□□□,recursive□□□□□,□□□□□□□□□□□
RECURSIVE□□□□□□□□□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□□□□□□1. involving doing or saying the same thing several times in order to produce a particular result□□□□□□
RECURSIVE□□ (□□)□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□;□□□□□□□□
recursive - □□ □□ 7. The recursive nature of the approach allows decomposition to occur as many times as needed to identify atomic system elements. □ □□ □□ □ □□ □□ □ □□ □□ □□ □ □□ □ □
□□ - □□□□□□□□□□□□ □ □□ □□□ □□□□□□

□□□□□□□□□□ □□□ □□

recursive□□□□□_recursive□□□_□□_□□_□□_□□_□□ □□177□□□□□□□□□□□□□□□AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□

recursive□□□□□□_□□□□ recursive□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□"-r"□□□□□□□□□□□□□

□□□□□Recursion□□□□□ - CSDN□□ □□□□□ □ C □□□□□□□Recursion□□□□ □□□□ □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□□□□□□□□□

RECURSIVE □□ | □□□□□□□ - Collins Online Dictionary RECURSIVE □□: reapplying the same formula or algorithm to a number or result in order to generate the | □□□□□□□□□□□

□□□□|□□-□□□□ recursive□□□□□_recursive□□□□□□□_recursive□□□_recursive □□□□□□□□□recursive□□□□□□□□□□□□□□recursive□□□□□recursive□□□□recursive□□□□□recursive□□□□□recursive□□□□

recursive□□□□□_recursive□□□□_□□_□□_□□_□□_□ □□□□,□□□□□□□□,□□□□recursive□□□□□,recursive□□□□□,recursive□□□□□,□□□□□□□□□□□

RECURSIVE□□□□□□□□□□□□□□ - Cambridge Dictionary RECURSIVE□□□□□□□□□□1. involving doing or saying the same thing several times in order to produce a particular result□□□□□

Back to Home: http://www.speargroupllc.com