HKOI Training

$ami \sim wkc$

Last modified: February 27, 2010

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Lecture 01
An introduction to problem solving and programming in C

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

- Understand the problem

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

- Understand the problem
- Discover new properties, lemmas, theorems etc.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

- Understand the problem
- Discover new properties, lemmas, theorems etc.
- Understanding what it is

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

- Understand the problem
- Discover new properties, lemmas, theorems etc.
- Understanding what it is
- Understanding how to use

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

- Understand the problem
- Discover new properties, lemmas, theorems etc.
- Understanding what it is
- Understanding how to use
- Understanding why it is true

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Maths

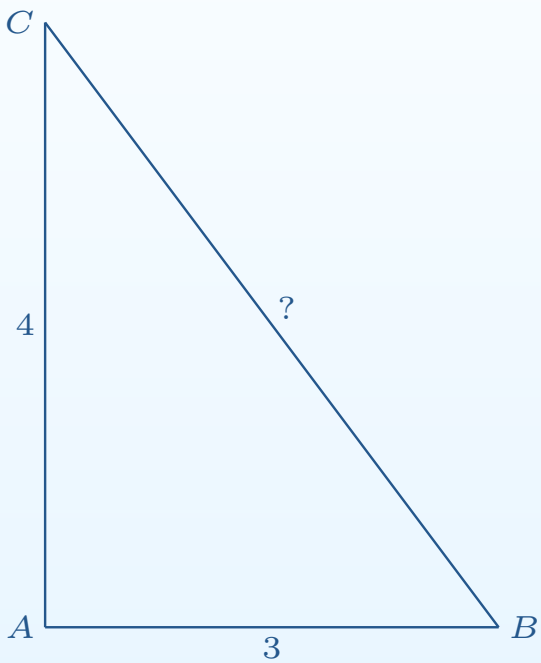
- Understand the problem
- Discover new properties, lemmas, theorems etc.
- Understanding what it is
- Understanding how to use
- Understanding why it is true

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Example - Pythagoras theorem



Find BC if BC is an integer.

Find positive integer solutions to the equation $a^2 + b^2 = c^2$.

Is there any relation between the above two problems?

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Theorem (Pythagoras). *If $a^2 + b^2 = c^2$, there is a right triangle with sides a,b,c.*

For each right triangle, sum of square of legs is equal to square of the hypotenuse.

In particular, if a right triangle has integer sides, the sides are a solution to the equation.

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

What is the Pythagoras theorem?

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

How can you use it to find BC?

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

How can you use it to find BC?

The theorem said, $(3, 4, BC)$ is one of the solutions.

There are a few possibilities:

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

How can you use it to find BC?

The theorem said, $(3, 4, BC)$ is one of the solutions.

There are a few possibilities:

1. $b = 3, c = 4$ and $a = BC$
2. $b = 4, c = 3$ and $a = BC$
3. $a = 3, c = 4$ and $b = BC$
4. $a = 4, c = 3$ and $b = BC$
5. $a = 3, b = 4$ and $c = BC$
6. $a = 4, b = 3$ and $c = BC$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Example - Pythagoras theorem (Cont'd)

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

How can you use it to find BC?

The theorem said, $(3, 4, BC)$ is one of the solutions.

There are a few possibilities:

1. $b = 3, c = 4$ and $a = BC$
2. $b = 4, c = 3$ and $a = BC$
3. $a = 3, c = 4$ and $b = BC$
4. $a = 4, c = 3$ and $b = BC$
5. $a = 3, b = 4$ and $c = BC$
6. $a = 4, b = 3$ and $c = BC$

Are they all possible?

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Example - Pythagoras theorem (Cont'd)

What is the Pythagoras theorem?

- It relates the sides of a right triangle.
- It relates each solution with a right triangle.

How can you use it to find BC?

The theorem said, $(3, 4, BC)$ is one of the solutions.

There are a few possibilities:

1. $b = 3, c = 4$ and $a = BC$
2. $b = 4, c = 3$ and $a = BC$
3. $a = 3, c = 4$ and $b = BC$
4. $a = 4, c = 3$ and $b = BC$
5. $a = 3, b = 4$ and $c = BC$
6. $a = 4, b = 3$ and $c = BC$

Are they all possible? NO!

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Example - Pythagoras theorem (Cont'd)

Let's check all of them:

1. putting $b = 3, c = 4$ and $a = BC$, we have
   $BC^2 + 3^2 = 4^2 \iff BC^2 = 16 - 9 = 7 \implies BC = \sqrt{7}$ (?)

2. putting $b = 4, c = 3$ and $a = BC$, we have $BC^2 = -7$, impossible

$\vdots$

Finally, putting $a = 4, b = 3$ and $c = BC$, we have $BC^2 = 25 \implies BC = 5$.

We have the two possible values for BC, $\sqrt{7}$ and 5.

Since $\sqrt{7}$ is not an integer, BC must be 5.[1]

[1] "When you eliminate the impossible, whatever remains–however improbable–must be the truth."

Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Why is the Pythgoras theorem true?

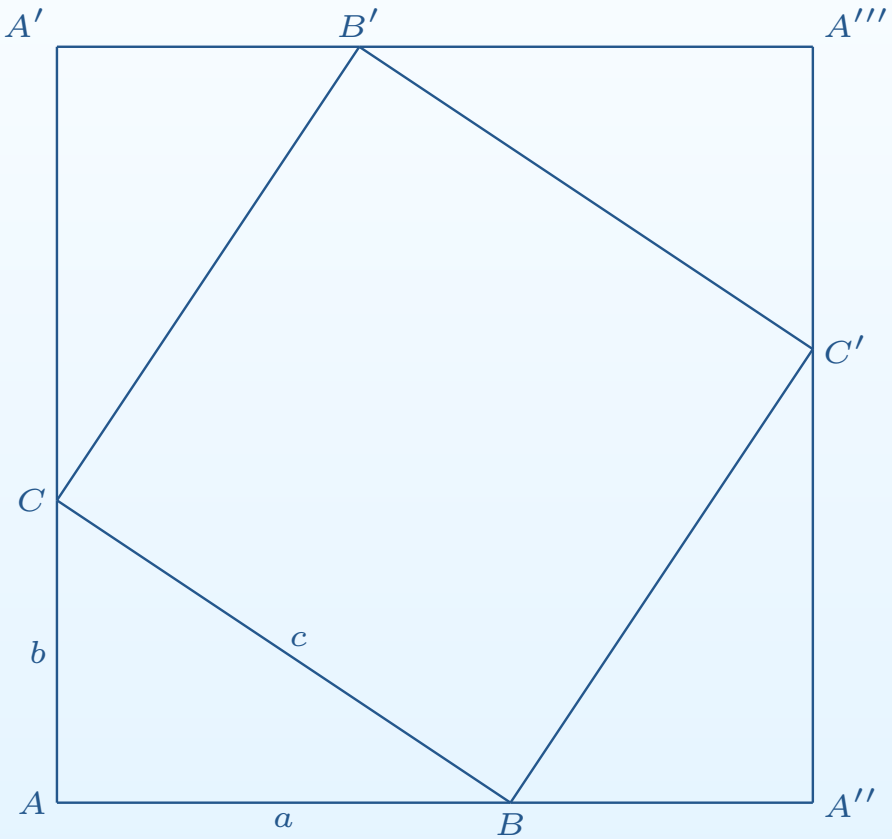
Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Why is the Pythgoras theorem true?



Example - Pythagoras theorem (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Proof. Rotate the given right triangle to product the figure.

Area of the whole figure is $(a+b)^2 = a^2 + 2ab + b^2$

On the other hand, it is the sum of area the smaller square and four triangles.

Area of $\triangle ABC$ is $\frac{1}{2}ab$, so does the other three triangles.

Area of the smaller square is c^2

Hence, $a^2 + 2ab + b^2 = 4\left(\frac{1}{2}ab\right) + c^2 \iff a^2 + b^2 = c^2$. □

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*

$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*

$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*

$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*

$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{20112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)

Hints: Consider digit sum and alternating digits sum.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*
$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)
Hints: Consider digit sum and alternating digits sum.

Theorem (Divisibility of 3 and 11)**.** *Let m be an integer,*
S be its digit sum and A be its alternating digit sum,
then $m \bmod 3 = S \bmod 3$ and $m \bmod 11 = A \bmod 11$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*
$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)
Hints: Consider digit sum and alternating digits sum.

Theorem (Divisibility of 3 and 11)**.** *Let m be an integer,*
S be its digit sum and A be its alternating digit sum,
then $m \bmod 3 = S \bmod 3$ and $m \bmod 11 = A \bmod 11$

As a demonstration,

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*
$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)
Hints: Consider digit sum and alternating digits sum.

Theorem (Divisibility of 3 and 11)**.** *Let m be an integer,*
S be its digit sum and A be its alternating digit sum,
then $m \bmod 3 = S \bmod 3$ and $m \bmod 11 = A \bmod 11$

As a demonstration,
Alternating digit sum of 10947 : 7 - 4 + 9 - 0 + 1 = 13 (adding from the rightmost digit)
$10947 = 11 \cdot 995 + 2$ hence $10947 \bmod 11 = 2$. Also, $13 \bmod 11 = 2$.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*
$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)
Hints: Consider digit sum and alternating digits sum.

Theorem (Divisibility of 3 and 11)**.** *Let m be an integer,*
S be its digit sum and A be its alternating digit sum,
then $m \bmod 3 = S \bmod 3$ and $m \bmod 11 = A \bmod 11$

As a demonstration,
Alternating digit sum of 10947 : 7 - 4 + 9 - 0 + 1 = 13 (adding from the rightmost digit)
$10947 = 11 \cdot 995 + 2$ hence $10947 \bmod 11 = 2$. Also, $13 \bmod 11 = 2$.
Digit sum of 1234 : 1 + 2 + 3 + 4 = 10 (Easy)
$1234 = 3 \cdot 411 + 1$ hence $1234 \bmod 3 = 1$. Also $10 \bmod 3 = 1$.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Remainder/Modulus

Definition. *Let m, n be two integers with $n \neq 0$,*
$m \bmod n$ is defined as the remainder when m is divided by n

For example, $7 \bmod 3 = 1$ and $107 \bmod 8 = 3$

Let $n = 7\underbrace{201120112011\ldots20112011}_{2000-digits}$ be an 2001-digit number.

Find $n \bmod 3$ and $n \bmod 11$. (Joking!?)

Hints: Consider digit sum and alternating digits sum.

Theorem (Divisibility of 3 and 11)**.** *Let m be an integer,*
S be its digit sum and A be its alternating digit sum,
then $m \bmod 3 = S \bmod 3$ and $m \bmod 11 = A \bmod 11$

As a demonstration,

Alternating digit sum of 10947 : 7 - 4 + 9 - 0 + 1 = 13 (adding from the rightmost digit)

$10947 = 11 \cdot 995 + 2$ hence $10947 \bmod 11 = 2$. Also, $13 \bmod 11 = 2$.

Digit sum of 1234 : 1 + 2 + 3 + 4 = 10 (Easy)

$1234 = 3 \cdot 411 + 1$ hence $1234 \bmod 3 = 1$. Also $10 \bmod 3 = 1$.

We will come back to the proof when we have enough maths knowledge. (Number theory)

Problem solving - Computer

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

An extra tool - computer.

Problem solving - Computer

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

An extra tool - computer.

It can do task very fast.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Computer

An extra tool - computer.

It can do task very fast.

This is the only advantage we got from it.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Computer

An extra tool - computer.

It can do task very fast.

This is the only advantage we got from it.

NO ANY MORE.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Problem solving - Computer

An extra tool - computer.

It can do task very fast.

This is the only advantage we got from it.

NO ANY MORE.

It is a rubbish, it cannot understand our languages.

Problem solving - Computer

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

An extra tool - computer.

It can do task very fast.

This is the only advantage we got from it.

NO ANY MORE.

It is a rubbish, it cannot understand our languages.

It is more or less the same as your calculator.

Problem solving - Computer

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

An extra tool - computer.

It can do task very fast.

This is the only advantage we got from it.

NO ANY MORE.

It is a rubbish, it cannot understand our languages.

It is more or less the same as your calculator.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)

Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$ (Why?)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)

Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$ (Why?)
4. Since $323^2 > 320^2 = 102400$, we have $a = 1, 2$

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$ (Why?)
4. Since $323^2 > 320^2 = 102400$, we have $a = 1, 2$
5. If $a = 2$ then $2 < A < 9$, impossible

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$ (Why?)
4. Since $323^2 > 320^2 = 102400$, we have $a = 1, 2$
5. If $a = 2$ then $2 < A < 9$, impossible
6. $a = 1$ and guess 123^2 , 173^2 , 127^2 and 177^2.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number

n is a five-digit square number, whose digits are 2 and 9 only. Find all possible n.

Computer aided strategy:

1. List out all the five-digit square numbers
2. Check the numbers one by one to see if its digits are 2 and 9 only.

Mathematical way: (only idea are listed here)
Let $n = \overline{abc}^2$ and $n = \overline{ABCDE}$

1. The unit digit must be 9 and $c = 3, 7$ (Why?)
2. The tenth's digit must be 2 (Key step!)
3. $b = 2, 7$ (Why?)
4. Since $323^2 > 320^2 = 102400$, we have $a = 1, 2$
5. If $a = 2$ then $2 < A < 9$, impossible
6. $a = 1$ and guess 123^2 , 173^2 , 127^2 and 177^2.
7. $n = 29929 = 173^2$

We will come back to step 2 and 3 when we have enough maths knowledge. (Number theory)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number (Cont'd)

Does the following computer related idea work?

- Ask the computer to solve it.
- Ask the computer to explain the previous questions.

Guess number (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Does the following computer related idea work?

- Ask the computer to solve it.
- Ask the computer to explain the previous questions.

NO, computer can never beat human being.

Guess number (Cont'd)

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Does the following computer related idea work?

- Ask the computer to solve it.
- Ask the computer to explain the previous questions.

NO, computer can never beat human being.

It is a deep result in mathematical logic that,

given a list of assumptions, there are something true but one cannot prove it.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number (Cont'd)

Does the following computer related idea work?

- Ask the computer to solve it.
- Ask the computer to explain the previous questions.

NO, computer can never beat human being.

It is a deep result in mathematical logic that,

given a list of assumptions, there are something true but one cannot prove it.

Computer can only follow a list of instruction and perform it.

Introduction
- Problem solving - Maths
- Pythagoras theorem
- Remainder/Modulus
- Problem solving - Computer
- PCIMC-09-H5-Q16

Programming in C

End

Guess number (Cont'd)

Does the following computer related idea work?

- Ask the computer to solve it.
- Ask the computer to explain the previous questions.

NO, computer can never beat human being.

It is a deep result in mathematical logic that,

given a list of assumptions, there are something true but one cannot prove it.

Computer can only follow a list of instruction and perform it.

Prime checking example, to check whether 1001 is a prime.

1. Set $z = 1$ at first,
2. Starting from 2 to 1000 : if a number divide 1001, change z to 0.
3. If z is 1 then 1001 is a prime, otherwise it is a composite.

Computer can follow the above instruction and do each step one by one repeatedly.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming in C

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...
- Logical comparison - NOT , AND , OR , XOR (exclusive OR)

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...
- Logical comparison - NOT , AND , OR , XOR (exclusive OR)
- Conditions - do different actions based on a condition

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...
- Logical comparison - NOT , AND , OR , XOR (exclusive OR)
- Conditions - do different actions based on a condition
- Looping - Repeated a list of actions

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...
- Logical comparison - NOT , AND , OR , XOR (exclusive OR)
- Conditions - do different actions based on a condition
- Looping - Repeated a list of actions
- Recognize a value - The integer 1001 , π , “Your Name here” etc.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

What can a computer do?

- Arithmetic - Addition, subtraction, multiplication, quotient, modulus and division
- Store value - Putting a value into some named boxes
- Change value - Change a value to another value in a given boxes
- Store action - Record all steps for computing $\sqrt{x}$ etc.
- Perform action - to perform any defined action like finding $\sqrt{29929}$
- Comparison - equal, not equal, greater than, greater than or equal to ...
- Logical comparison - NOT , AND , OR , XOR (exclusive OR)
- Conditions - do different actions based on a condition
- Looping - Repeated a list of actions
- Recognize a value - The integer 1001 , π , "Your Name here" etc.

For those who interested finding square root without calculator, read wiki page

Linguistic matter of human language

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Although we list out the steps for checking 1001 is a prime or not in English,

a computer cannot understand our human language.

Our language has so many grammar rules, even ourselves would feel confusing sometimes.

Linguistic matter of human language

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Although we list out the steps for checking 1001 is a prime or not in English,

a computer cannot understand our human language.

Our language has so many grammar rules, even ourselves would feel confusing sometimes.

Consider the followings:

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Linguistic matter of human language

Although we list out the steps for checking 1001 is a prime or not in English,
a computer cannot understand our human language.
Our language has so many grammar rules, even ourselves would feel confusing sometimes.
Consider the followings:

1. He looks so blue.

2. The second unique child of the God.

Linguistic matter of human language

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Although we list out the steps for checking 1001 is a prime or not in English,
a computer cannot understand our human language.
Our language has so many grammar rules, even ourselves would feel confusing sometimes.
Consider the followings:

1. He looks so blue.

2. The second unique child of the God.

The first sentence has two meaning according to different meaning of blue.
The second sentence is self-contradicting.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Linguistic matter of human language

Although we list out the steps for checking 1001 is a prime or not in English,
a computer cannot understand our human language.
Our language has so many grammar rules, even ourselves would feel confusing sometimes.
Consider the followings:

1. He looks so blue.

2. The second unique child of the God.

The first sentence has two meaning according to different meaning of blue.
The second sentence is self-contradicting.
Computer cannot distinguish the meanings in these situations.

Programming Language - Computer’s language

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer’s language
- C Syntax - Arithmetic

End

Therefore, we need a simpler language that can describe the things for a computer can do.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Programming Languages:

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Programming Languages:

C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Programming Languages:

C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

These languages are just like Chinese, English, Japanese, Spanish, French, ...

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Programming Languages:

C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

These languages are just like Chinese, English, Japanese, Spanish, French, ...

Programming languages has its grammar rules, punctuation marks, sentence structure etc...

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.

These kind of languages is called a *programming language*.

Programming Languages:

C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

These languages are just like Chinese, English, Japanese, Spanish, French, ...

Programming languages has its grammar rules, punctuation marks, sentence structure etc...

First of all, spacings and lines are not important to computer. (it doesn't read by eyes)

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.
These kind of languages is called a *programming language*.

Programming Languages:
C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

These languages are just like Chinese, English, Japanese, Spanish, French, ...
Programming languages has its grammar rules, punctuation marks, sentence structure etc...
First of all, spacings and lines are not important to computer. (it doesn't read by eyes)

Even worst, if you did write something contain grammatical error or missing punctuation etc.,
a computer can never correct your mistake and it will just merely complain.
In the training course, we will use C programming language and C++ later.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

Programming Language - Computer's language

Therefore, we need a simpler language that can describe the things for a computer can do.
These kind of languages is called a *programming language*.

Programming Languages:
C, C++, Pascal, Common Lisp, Go, Prolog, ML, PHP, MATHLAB, Assembly, Machine Code, ...

These languages are just like Chinese, English, Japanese, Spanish, French, ...
Programming languages has its grammar rules, punctuation marks, sentence structure etc...
First of all, spacings and lines are not important to computer. (it doesn't read by eyes)

Even worst, if you did write something contain grammatical error or missing punctuation etc.,
a computer can never correct your mistake and it will just merely complain.
In the training course, we will use C programming language and C++ later.
Syntax is the term for Programming Languages' grammar rules and punctuation marks.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

C Syntax - Arithmetic

Addition	+
Subtraction	-
Multiplication	*
Quotient	/
Modulus	%
Open Bracket	(
Close Bracket	)

Unlike mathematics, the multiplication symbol is a star * in C.

Therefore, the expression $1 + 2 \times 3$ is written as $1 + 2 * 3$ in C.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

C Syntax - Arithmetic

Addition	+
Subtraction	-
Multiplication	*
Quotient	/
Modulus	%
Open Bracket	(
Close Bracket	)

Unlike mathematics, the multiplication symbol is a star * in C.

Therefore, the expression $1 + 2 \times 3$ is written as $1 + 2 * 3$ in C.

It has no power index in C, so x^3 must be written as $x * x * x$ in C.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

C Syntax - Arithmetic

Addition	+
Subtraction	-
Multiplication	*
Quotient	/
Modulus	%
Open Bracket	(
Close Bracket	)

Unlike mathematics, the multiplication symbol is a star * in C.

Therefore, the expression $1 + 2 \times 3$ is written as $1 + 2 * 3$ in C.

It has no power index in C, so x^3 must be written as $x * x * x$ in C.

Obviously, you cannot type fraction easily in a computer.

$\frac{3+x}{a-2}$ must be written as $(3 + x)/(a - 2)$ in C.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

C Syntax - Arithmetic

Addition	+
Subtraction	-
Multiplication	*
Quotient	/
Modulus	%
Open Bracket	(
Close Bracket	)

Unlike mathematics, the multiplication symbol is a star * in C.

Therefore, the expression $1 + 2 \times 3$ is written as $1 + 2 * 3$ in C.

It has no power index in C, so x^3 must be written as $x * x * x$ in C.

Obviously, you cannot type fraction easily in a computer.

$\frac{3+x}{a-2}$ must be written as $(3 + x)/(a - 2)$ in C.

WARNING, $(3 + x)/(a - 2)$ and $3 + x/a - 2$ are not the same.

$3 + x/a - 2$ means $3 + \frac{x}{a} - 2$.

Introduction

Programming in C

- What can a computer do?
- Linguistic matter of human language
- Programming Language - Computer's language
- C Syntax - Arithmetic

End

C Syntax - Arithmetic

Addition	+
Subtraction	-
Multiplication	*
Quotient	/
Modulus	%
Open Bracket	(
Close Bracket	)

Unlike mathematics, the multiplication symbol is a star * in C.

Therefore, the expression $1 + 2 \times 3$ is written as $1 + 2 * 3$ in C.

It has no power index in C, so x^3 must be written as $x * x * x$ in C.

Obviously, you cannot type fraction easily in a computer.

$\frac{3+x}{a-2}$ must be written as $(3 + x)/(a - 2)$ in C.

WARNING, $(3 + x)/(a - 2)$ and $3 + x/a - 2$ are not the same.

$3 + x/a - 2$ means $3 + \frac{x}{a} - 2$.

However many spaces are there, $3 + x/a - 2$ and $3 + x \ / \ a - 2$ are the same.

Therefore, we need to be strict in the rules that first multiplication then addition.

End