

บทที่ 8

ฟังก์ชัน (Function)





Function

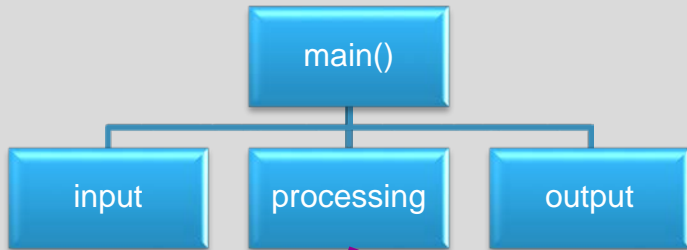
- ❖ ฟังก์ชัน คือ ชุดคำสั่งที่มีชื่อกำกับ สามารถเรียกให้ทำงานได้
หลายๆครั้ง
- ❖ ใช้แบ่งส่วนของโค้ดออกเป็น Module
- ❖ ใช้แยกส่วนของโค้ดที่ถูกเรียกใช้บ่อยๆ ออกมา เพื่อ reuse หรือใช้ร่วมกับโปรแกรมอื่นๆได้



การ reuse ส่วนของโค้ด

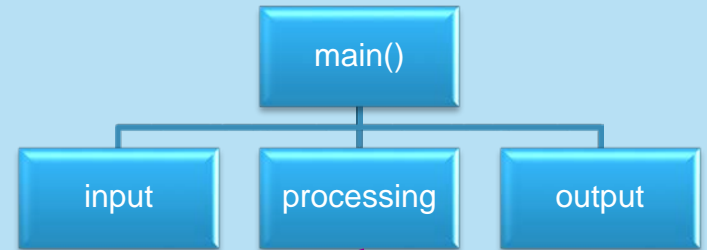
โปรแกรมหาปริมาตรทรงกลม

การประมวล คือ $\frac{4}{3}\pi r^3$



โปรแกรมหาปริมาตรทรงกระบอก

การประมวล คือ $\pi r^2 h$





ประเภทของฟังก์ชัน

❖ **Standard Library Function** คือ ฟังก์ชันมาตรฐานที่ติดมากับตัวแปลภาษาซี เช่น

- ฟังก์ชัน printf(), scanf() มาจาก Library stdio.h
- ฟังก์ชัน strstr(), strcmp() มาจาก Library string.h

❖ **User-define Function** คือ ฟังก์ชันที่เขียนขึ้นมาใหม่เพื่อรองรับการทำงานตามความต้องการ



รูปแบบฟังก์ชัน

Header

ชนิดข้อมูลที่ส่งกลับ ชื่อฟังก์ชัน(ชนิดข้อมูลที่รับเข้า ตัวแปรรับเข้า) {

Body

ชุดคำสั่งต่างๆ

Return Statement

return ค่าส่งกลับ;

}



ฟังก์ชันรูปแบบต่างๆ



แบบไม่รับและไม่ส่งข้อมูลกลับ

printMenu

แบบรับข้อมูล แต่ไม่ส่งข้อมูลกลับ

name

printGreeting

แบบรับและส่งข้อมูลกลับ

num1

num2

num3

findAverage

ค่าเฉลี่ย



แบบไม่รับและไม่ส่งข้อมูลกลับ

printMenu

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

ไม่มีการรับข้อมูลเข้า

ไม่มีการส่งข้อมูลกลับ

```
void printMenu() {  
    printf("1. Add customer\n");  
    printf("2. Edit customer\n");  
    printf("3. Delete customer\n");  
    printf("Please select menu: ");  
}
```

```
void main() {
```

เรียกใช้ฟังก์ชัน

```
    printMenu();  
}
```



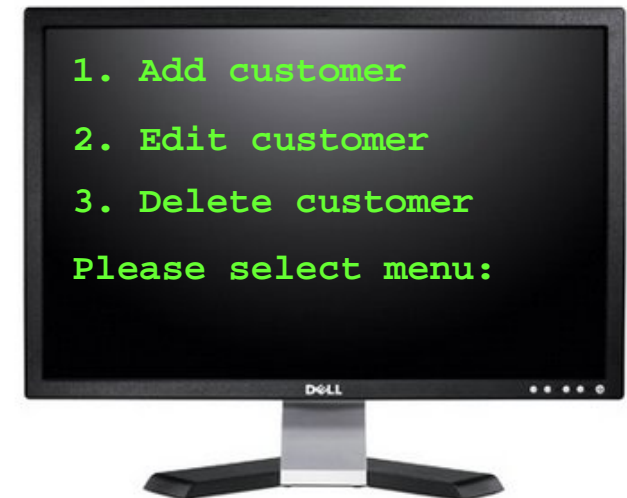
แบบไม่รับและไม่ส่งข้อมูลกลับ

printMenu

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

```
void printMenu() {  
    printf("1. Add customer\n");  
    printf("2. Edit customer\n");  
    printf("3. Delete customer\n");  
    printf("Please select menu: ");  
}
```

```
void main() {  
    printMenu();  
}
```





แบบรับข้อมูล แต่ไม่ส่งข้อมูลกลับ

name →

printGreeting

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

รับข้อมูลเข้า

ไม่มีการส่งข้อมูลกลับ

```
void printGreeting(char name[]) {  
    printf("Hello %s\n", name);  
}
```

```
void main() {
```

```
    printGreeting("Celine Dion");
```

เรียกใช้ฟังก์ชัน

```
    char myName[] = "Mariah Carey";
```

```
    printGreeting(myName);
```

ส่งข้อมูลไปยังฟังก์ชัน

```
}
```



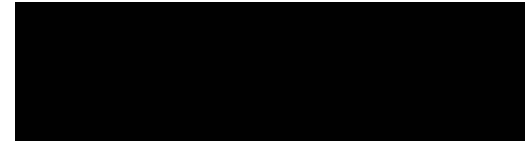
แบบรับข้อมูล แต่ไม่ส่งข้อมูลกลับ

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

```
void printGreeting(char name[]) {  
    printf("Hello %s\n", name);  
}
```

```
void main() {  
    printGreeting("Celine Dion");
```

```
    char myName[] = "Mariah Carey";  
    printGreeting(myName);  
}
```



name []

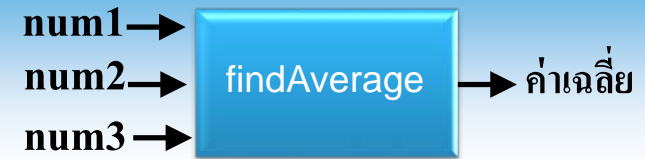


myName []





แบบรับและส่งข้อมูลกลับ



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

ชนิดข้อมูลที่ส่งกลับ

```
float findAverage(int num1, int num2, int num3) {  
    int sum = num1 + num2 + num3;  
    float avg;  
    avg = sum/3.0;  
    return avg; }  
                •———— ข้อมูลที่ส่งกลับ
```

รับข้อมูลเข้า

```
void main() {  
    float average;  
    average = findAverage(3, 7, 6);  
    printf("%.2f", average);  
}
```

เรียกใช้ฟังก์ชัน

ส่งข้อมูลไปยังฟังก์ชัน



แบบรับและส่งข้อมูลกลับ

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



```
float findAverage(int num1, int num2, int num3) {
```

```
int sum = num1 + num2 + num3;
```

```
float avg;
```

```
avg = sum/3.0;
```

```
return avg;
```

```
}
```



```
void main() {
```

```
float average;
```

```
average = findAverage(3, 7, 6);
```

```
printf("%.2f", average);
```

```
}
```





กิจกรรม

จงเขียนฟังก์ชันในการหาค่า Absolute และเรียกใช้ฟังก์ชัน



```
#include <stdio.h>

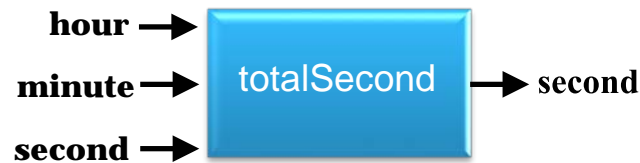
_____ findAbsolute ( _____ ) {
    int absolute;
    if (number >=0)
        absolute = number;
    else
        absolute = -number;
    return _____;
}

void main(){
    int ab = findAbsolute(-15);
    printf("%d", ab);
}
```



กิจกรรม

- ❖ จงเขียนฟังก์ชันในการหาค่าวินาที เมื่อมีการกำหนดข้อมูลเข้าเป็น จำนวนชั่วโมง นาที และวินาที



```
#include <stdio.h>

_____ totalSecond ( _____ , _____ , _____ ) {
    _____
    _____
}

void main(){
    printf("%d", totalSecond(1, 2, 6));
}
```



Parameter และ Argument

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

```
float findAverage(int num1, int num2, int num3) {  
    int sum = num1 + num2 + num3;  
    float avg;  
    avg = sum/3.0;  
    return avg;  
}
```

ตัวแปรที่ฟังก์ชันใช้รับค่า
เช่น num1, num2, num3

เรียกว่า **Parameter**

```
void main() {  
    float average;  
    int someValue = 15;  
    average = findAverage(3, someValue, 6);  
    printf("%.2f", average);  
}
```

ตัวแปรที่ส่งไปให้ฟังก์ชัน เช่น someValue
และค่าที่ส่งไปให้ฟังก์ชัน เช่น 3 และ 6

เรียกว่า **Argument**



Function Prototype

```
#include <stdio.h>

float findAverage(int num1, int num2, int num3) {
    int sum = num1 + num2 + num3;
    float avg;
    avg = sum/3.0;
    return avg;
}

void main() {
    float average;
    average = findAverage(3, 7, 6);
    printf("%.2f", average);
}
```

กรณีฟังก์ชันอยู่เหนือ main() **ไม่ต้องเขียน**

Function Prototype

```
#include <stdio.h>

float findAverage(int, int, int);

void main() {
    float average;
    average = findAverage(3, 7, 6);
    printf("%.2f", average);
}

float findAverage(int num1, int num2, int num3) {
    int sum = num1 + num2 + num3;
    float avg;
    avg = sum/3.0;
    return avg;
}
```

กรณีฟังก์ชันอยู่ใต้ main() **จะต้องเขียน**

Function Prototype

Function Prototype คือ ส่วนหัวของฟังก์ชัน เพื่อบอกให้ main() รู้จักฟังก์ชันนั้น เนื่องจาก main() จะรู้จักฟังก์ชันที่อยู่เหนือมันเท่านั้น



ตัวแปรชนิด Local

ขอบเขตของตัวแปร sum ซึ่งเป็น Local Variable จะเห็นได้เฉพาะในฟังก์ชันที่ประกาศเท่านั้น

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

```
void add();
```

```
void main()
```

```
{  
    int sum = 0;  
    sum = sum + 2;  
    printf("%d", sum);  
}
```

```
void add()
```

```
{  
    sum = sum + 4;  
    printf("%d", sum);  
}
```



ไม่สามารถใช้ Local Variable
sum ในฟังก์ชันอื่นได้



ตัวแปรชนิด Global

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

```
int tax = 7; ●———— Global Variable
```

```
● int calTax()  
{  
    int a = 4;  
    tax = tax + a;  
    return tax;  
}
```

```
void main()  
{  
    printf("Tax: %d", tax);  
● }
```

ขอบเขตของตัวแปร tax
สามารถเห็นได้ในทุกฟังก์ชัน



การดึงส่วนของโค้ดไปเป็นฟังก์ชัน

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

```
void say();
```

```
void main() {  
    printf("Hello World\n");  
    printf("Hi World\n");  
    printf("Hey World\n");  
    printf("What's up World\n");  
    printf("Bye World\n");  
}
```

```
void say()
```

```
{
```

```
}
```



การดึงส่วนของโค้ดไปเป็นฟังก์ชัน

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

```
float toDollar(int);
```

```
void main() {
```

```
    int baht;
```

```
    float dollar;
```

```
    printf("Enter your money in baht: ");
```

```
    scanf("%d", &baht);
```

```
    dollar = toDollar(baht);
```

```
    printf("%d Thai Baht equals %.2f US Dollar", baht, dollar);
```

```
}
```

```
float toDollar(int baht) {
```

```
    float result =
```

```
    return result;
```

```
}
```



กิจกรรม

จากโปรแกรมหาพื้นที่สามเหลี่ยมต่อไปนี้ ให้ทำการแยกส่วนคำนวณพื้นที่สามเหลี่ยมออกมาเป็นฟังก์ชัน

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

```
void main() {  
    float area;  
    int base, height;  
    printf("Please enter base> ");  
    scanf("%d", &base);  
    printf("Please enter height> ");  
    scanf("%d", &height);  
  
    area = 1 / 2.0 * base * height;  
    printf("Area is %f", area);  
}
```

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

```
void main() {  
    float area;  
    int base, height;  
    printf("Please enter base> ");  
    scanf("%d", &base);  
    printf("Please enter height> ");  
    scanf("%d", &height);  
  
    area = _____  
    printf("Area is %f", area);  
}
```



กิจกรรม

จงเขียนฟังก์ชัน `printLine` ที่ทำหน้าที่ในการพิมพ์ตัวอักขระใดๆ จำนวน n ตัว โดยมีรูปแบบฟังก์ชัน ดังนี้

```
void printLine(char c, int n) { ... }
```

หลังจากนั้นให้เขียนฟังก์ชัน `main` เพื่อรับค่าและและส่งไปทำงานที่ฟังก์ชัน `printLine` โปรแกรมจะทำงานไปเรื่อยๆ จนกว่าจะใส่อักขระเป็น `q` ตัวอย่างผลการทำงานแสดงดังด้านล่างนี้

ตัวอย่างหน้าจอโปรแกรม

ขั้นตอนการเขียนโปรแกรม

1. ออกแบบฟังก์ชัน
2. ทดสอบเรียกใช้ฟังก์ชัน

```
Input character: p ↵  
Input number of character: 8 ↵  
pppppppp  
  
Input character: * ↵  
Input number of character: 10 ↵  
*****  
  
Input character: q ↵  
Bye!
```



ออกแบบฟังก์ชัน

```
void printLine(char c, int n) {  
    int i;  
    for (i=0; i<n; i++)  
        printf("%c", c);  
    printf("\n");  
}
```



ทดสอบเรียกใช้ฟังก์ชัน

```
printf("Input character:");  
scanf(" %c", &c);  
printf("Input length:");  
scanf("%d", &length);  
println(c, length);
```




กิจกรรม

จงเขียนฟังก์ชัน myPow เพื่อใช้ในการหาตัวเลขยกกำลังด้วยตนเอง (แทนการใช้ฟังก์ชันจาก math.h) โดยมีพารามิเตอร์เป็นเลขฐาน และเลขยกกำลัง รูปแบบฟังก์ชัน คือ

```
float myPow(float base, float power) { ... }
```

หลังจากนั้นให้เขียนฟังก์ชัน main เพื่อนำฟังก์ชัน myPow ไปใช้คำนวณปริมาตรทรงกลม ซึ่งมีสูตรดังนี้

$$\text{สูตรปริมาตรทรงกลม} = \frac{4}{3} \pi r^3$$

โดยให้รับค่าจากผู้ใ้ แล้วเขียนนิพจน์ที่มีการใช้ฟังก์ชัน myPow และแสดงผลลัพธ์

ตัวอย่างหน้าจอโปรแกรม

```
Input radius: 2 ↵  
Volume of a sphere is 33.49
```



ออกแบบฟังก์ชัน

```
float myPow(float base, float power) {  
    int i;  
    float result = 1;  
    for (i=0; i<power; i++)  
        result *= base;  
    return result;  
}
```



ทดสอบเรียกใช้ฟังก์ชัน

```
volume = 4 / 3.0 * 3.14 * myPow(radius, 3);
```



กิจกรรม

จงเขียนฟังก์ชันที่ช่วยในการแปลงปี พ.ศ. (Buddhist Era) และปี ค.ศ. (Christian Era)
2 ฟังก์ชัน ดังนี้

1. แปลงปี พ.ศ. เป็นปี ค.ศ. มีรูปแบบฟังก์ชัน คือ

```
int toChristian(int year) { .... }
```

2. แปลงปี ค.ศ. เป็นปี พ.ศ. มีต้นแบบฟังก์ชัน คือ

```
int toBuddhist(int year) { .... }
```

หลังจากนั้นให้ทดสอบฟังก์ชัน โดยเรียกใช้ในฟังก์ชัน main

ตัวอย่างหน้าจอโปรแกรม

```
Enter Buddhist Year: 2558 ↵  
Christian Year is 2015
```

```
Enter Christian Year: 2011 ↵  
Buddhist Year is 2554
```



กิจกรรม

จงเขียนฟังก์ชันเพื่อหาค่าสมการ $f(x) = 2x^2 + x - 21$ โดย x คือค่าที่รับจากผู้ใช้งาน มีรูปแบบฟังก์ชัน คือ

```
int f(int x) { .... }
```

หลังจากนั้นให้ทดสอบฟังก์ชัน โดยเรียกใช้ในฟังก์ชัน main

ตัวอย่างหน้าจอโปรแกรม

```
x = 4 ↵  
f(4) = 15
```

Test case:

Input	Output
x = 0	f(0) = -21
x = 1	f(1) = -18
x = 2	f(2) = -11
x = 3	f(3) = 0
x = 4	f(4) = 15
x = 5	f(5) = 34



กิจกรรม

จงเขียนฟังก์ชันเพื่อหาค่า Factorial ของ n โดย n คือค่าที่รับจากผู้ใช้ มีรูปแบบฟังก์ชัน คือ

```
int factorial(int n) { .... }
```

หลังจากนั้นให้ทดสอบฟังก์ชัน โดยเรียกใช้ในฟังก์ชัน main

ตัวอย่างหน้าจอโปรแกรม

```
n = 5 ↵  
5! = 120
```

Test case:

Input	Output
5	120
3	6
0	1



ฟังก์ชันมาตรฐานในภาษาซี

❖ math.h

- `sqrt(x)` – หาค่า \sqrt{x}
- `pow(x, y)` – หาค่า x^y
- `floor(x)` – ปัดเศษตัวแปร `x` ลง

❖ stdlib.h

- `atoi(s)` – แปลง String เป็น int
- `atof(s)` – แปลง String เป็น float
- `system(char command[])` สั่งให้รันคำสั่งบน DOS Command Line



การเรียกใช้ฟังก์ชันมาตรฐาน

ชื่อฟังก์ชันมาตรฐาน

```
result = sqrt ( x );
```

ตัวแปรสำหรับเก็บผลลัพธ์

ค่าที่ต้องการคำนวณ
จะเป็นตัวแปรหรือตัวเลขก็ได้



ตัวอย่าง

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

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

ระบุชื่อ Library

```
void main() {  
    float x, result;  
    printf("Enter number: ");  
    scanf("%f", &x);  
  
    result = sqrt(x);  
    printf("%f", result);  
}
```

เรียกใช้ฟังก์ชันมาตรฐาน



ตัวอย่าง

```
#include <stdio.h>
#include <math.h>
void main() {
    int x, y;
    float result;
    printf("Enter x and y: ");
    scanf("%d %d", &x, &y);

    result = sqrt(pow(x,2) + pow(y,2));
    printf("%f", result);
}
```

คำนวณผลลัพธ์จากสูตร

$$\sqrt{x^2 + y^2}$$



ไลบรารี stdio.h

ฟังก์ชัน	หน้าที่
<code>int getchar()</code>	อ่านข้อมูล 1 อักขระจากอุปกรณ์รับข้อมูลมาตรฐาน
<code>char *gets(const char *buffer)</code>	อ่านข้อมูลสตริงจากอุปกรณ์รับข้อมูลมาตรฐานมาเก็บในตัวแปร <code>buffer</code> จนกว่าจะมีการขึ้นบรรทัดใหม่หากมีความผิดพลาดจะคืนค่า <code>NULL</code>
<code>void printf(const char *format, ...)</code>	ใช้แสดงผลข้อมูลทางอุปกรณ์แสดงผลมาตรฐานในรูปแบบที่กำหนด
<code>void putchar(int c)</code>	ส่งอักขระไปแสดงผลทางอุปกรณ์แสดงผลมาตรฐาน
<code>void puts(const char *buffer)</code>	ส่งข้อมูลสตริงไปแสดงผลทางอุปกรณ์แสดงผลมาตรฐาน
<code>void scanf(const char *format,...)</code>	ใช้อ่านข้อมูลทางอุปกรณ์รับข้อมูลมาตรฐานในรูปแบบที่กำหนด



ไลบรารี stdlib.h

ฟังก์ชัน	หน้าที่
<code>int abs(int n)</code>	หาค่าสัมบูรณ์ของค่า <code>n</code>
<code>float atof(const char *s)</code>	แปลงค่าสตริง <code>s</code> เป็นค่าข้อมูลชนิด <code>float</code>
<code>int atoi(const char *s)</code>	แปลงค่าสตริง <code>s</code> เป็นค่าข้อมูลชนิด <code>int</code>
<code>long atol(const char *s)</code>	แปลงค่าสตริง <code>s</code> เป็นค่าข้อมูลชนิด <code>long</code>
<code>int rand(void)</code>	คืน ค่าเลขจำนวนเต็มที่สุ่มขึ้นมาอยู่ในช่วง 0 ถึง <code>RAND_MAX</code> ซึ่งไม่เกิน 32767



ไลบรารี math.h

ฟังก์ชัน	หน้าที่
$\text{acos}(x)$	หาค่า Arc Cosine ของ x
$\text{asin}(x)$	หาค่า Arc Sine ของ x
$\text{atan}(x)$	หาค่า Arc Tangent ของ x
$\text{atan2}(x,y)$	หาค่า Arc Tangent ของ x / y
$\text{ceil}(x)$	หาค่าจำนวนจริง x ที่มีการปัดเศษขึ้นทั้งหมด
$\text{cos}(x)$	หาค่า Cosine ของ x
$\text{exp}(x)$	หาค่าเอ็กโปเนนเชียล (Exponential) ของ x
$\text{floor}(x)$	หาค่าจำนวนจริง x ที่มีการปัดเศษลงทั้งหมด
$\text{log}(x)$	หาค่า Logarithm ของ x
$\text{log10}(x)$	หาค่า Logarithm ฐาน 10 ของ x
$\text{pow}(x, y)$	หาค่า x ยกกำลัง y
$\text{sin}(x)$	หาค่า Sine ของ x
$\text{sqrt}(x)$	หาค่า Square Root ของ x
$\text{tan}(x)$	หาค่า Tangent ของ x



ไลบรารี ctype.h

ฟังก์ชัน	หน้าที่
int isalnum(c)	ตรวจสอบตัวอักษรนั้นเป็นตัวอักษร 0-9 หรือ a-z หรือ A-Z หรือไม่ ถ้าไม่ใช่คืนค่า 0 ถ้าใช่คืนค่าที่ไม่ใช่ 0
int isalpha(c)	ตรวจสอบตัวอักษรนั้นเป็นตัวอักษร a-z หรือ A-Z หรือไม่ ถ้าไม่ใช่คืนค่า 0 ถ้าใช่คืนค่าคงที่ไม่ใช่ 0
int isdigit(c)	ตรวจสอบตัวอักษรนั้นเป็นตัวอักษร 0-9 หรือไม่ ถ้าไม่ใช่คืนค่า 0 ถ้าใช่คืนค่าที่ไม่ใช่ 0
int islower(c)	ตรวจสอบตัวอักษรนั้นเป็นตัวอักษร a-z หรือไม่ ถ้าไม่ใช่คืนค่า 0 ถ้าใช่คืนค่าที่ไม่ใช่ 0
int isupper(c)	ตรวจสอบตัวอักษรนั้นเป็นตัวอักษร A-Z หรือไม่ ถ้าไม่ใช่คืนค่า 0 ถ้าใช่คืนค่าที่ไม่ใช่ 0
int tolower(c)	แปลงค่าอักขระปัจจุบันให้เป็นอักขระเดียวกันแต่เป็นตัวพิมพ์เล็ก
int toupper(c)	แปลงค่าอักขระปัจจุบันให้เป็นอักขระเดียวกันแต่เป็นตัวพิมพ์ใหญ่



คู่มือการใช้งานฟังก์ชันมาตรฐานภาษาซี

<http://code-reference.com/c>

<http://www.cplusplus.com/reference/clibrary/>

http://www.tutorialspoint.com/c_standard_library



กิจกรรม

จงเขียนโปรแกรมคำนวณระยะห่างระหว่างจุด 2 จุด คือ (x_1, y_1) และ (x_2, y_2) โดยใช้สูตร

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ตัวอย่างหน้าจอโปรแกรม

```
Enter x1 and y1 (separate by space): 5 6 ↵  
Enter x2 and y2 (separate by space): 2 3 ↵  
distance = 4.24
```