

ฟังก์ชันมาตรฐาน

ฟังก์ชันมาตรฐานเป็นฟังก์ชันที่บริษัทผู้ผลิต c เขียนขึ้น บางครั้งเรียกไลบรารีฟังก์ชัน

1. ฟังก์ชันทางคณิตศาสตร์ (mathematic functions)

เป็นฟังก์ชันใช้คำนวณทางคณิตศาสตร์ ก่อนใช้ฟังก์ชันนี้ จะต้องใช้คำสั่ง `#include <math.h>` แทรกอยู่ตอนต้นโปรแกรม ตัวแปรที่ใช้ฟังก์ชันประเภทนี้ จะต้องมีชนิด (type) เป็น `double` ผลลัพธ์ที่ได้ของฟังก์ชัน ค่าส่งกลับของข้อมูลเป็น `double`

ฟังก์ชันทางคณิตศาสตร์ที่ควรรทราบ

acos(x) asin(x) atan(x) sin(x)

cos(x) tan(x) sqrt(x) exp(x)

pow(x,y) log(x) log10(x) ceil(x)

floor(x) fabs(x)

1. ฟังก์ชัน $\text{acos}(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า arc cosine ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\text{acos}(x);$

2. ฟังก์ชัน $\text{asin}(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า arc sine ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\text{asin}(x);$

3. ฟังก์ชัน $\text{atan}(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า arc tan ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\text{atan}(x);$

4. ฟังก์ชัน $\text{sin}(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า sine ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\text{sin}(x);$

5. ฟังก์ชัน $\cos(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า **cosine** ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\cos(x)$;

6. ฟังก์ชัน $\tan(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า **tan** ของ x โดย x เป็นค่ามุมในหน่วยเรเดียน (radian)

$\tan(x)$;

ทดลองเขียนฟังก์ชัน

```
/* math1.c */
#include<stdio.h>
#include<math.h>
void main(void)
{
    double r, pi=3.141592654;
    r=pi/180;
    printf("%f\n",asin(r));
    printf("%f\n",acos(r));
    printf("%f\n",atan(r));
    printf("%f\n",sin(r));
    printf("%f\n",cos(r));
    printf("%f\n",tan(r));
}
```

7. ฟังก์ชัน $\text{sqrt}(x)$

เป็นฟังก์ชันที่ใช้หาค่า รากที่ 2 (square root) ของ ค่าคงที่ หรือ ตัวแปร X

$\text{sqrt}(x);$

8. ฟังก์ชัน $\text{exp}(x)$

เป็นฟังก์ชันที่ใช้คำนวณหาค่า e^x โดย X เป็นค่าคงที่หรือตัวแปรที่จะใช้เป็นค่ายกกำลังของ e โดยที่ e มีค่าประมาณ 2.718282

$\text{exp}(x);$

7. ฟังก์ชัน `pow(x,y)`

เป็นฟังก์ชันที่ใช้หาค่า X ยกกำลัง y

`pow(x,y);`


```
/* math2.c */  
#include<stdio.h>  
#include<math.h>  
#include<conio.h>  
  
void main(void)  
{  
    double a = 5.0, b=2.0, c=16.0;  
    printf("%.4f\n",pow(a,b));  
    printf("%.4f\n",sqrt(c));  
    printf("%.4f\n",exp(b));  
}
```

10. ฟังก์ชัน $\log(x)$

เป็นฟังก์ชันที่ใช้หาค่า \log ฐาน n (natural logarithm) ของ ค่าคงที่หรือ ตัวแปร X โดยที่ X เป็นค่าคงที่หรือตัวแปรที่มีค่าเป็นลบไม่ได้

$$\log(x);$$

11. ฟังก์ชัน $\log_{10}(x)$

เป็นฟังก์ชันที่ใช้หาค่า \log ฐาน 10 ของค่าคงที่หรือตัวแปร X โดยที่ X เป็นค่าคงที่หรือตัวแปรที่มีค่าลบไม่ได้

$$\log_{10}(x);$$

```
/* math3.c */  
#include<stdio.h>  
#include<math.h>  
#include<conio.h>  
void main(void)  
{  
    double x=10.0, y=3.0;  
    printf("%.4f\n",log(y));  
  
    printf("%.4f\n",log10(x));  
}
```

12. ฟังก์ชัน $\text{ceil}(x)$

เป็นฟังก์ชันที่ใช้ในการปัดเศษทศนิยมขึ้นของตัวแปร X ถ้า X เป็นตัวเลขทศนิยม แต่ถ้า X เป็นเลขจำนวนเต็มจะไม่มี การปัดเศษทศนิยม

$\text{ceil}(x);$

13. ฟังก์ชัน $\text{floor}(x)$

เป็นฟังก์ชันที่ใช้ในการตัดเศษทศนิยมทิ้งของตัวแปร X ถ้า X เป็นตัวเลขจำนวนทศนิยม แต่ถ้า X เป็นเลขจำนวนเต็มจะไม่มี การตัดเศษทศนิยมทิ้ง

$\text{floor}(x);$

```
/* math4.c */
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main(void)
{
    printf("%.4f\n",ceil(4.6782));
    printf("%.4f\n",ceil(-9.6545));
    printf("%.4f\n",ceil(150));
    printf("%.4f\n",floor(6.8765));
    printf("%.4f\n",floor(-2.4563));
    printf("%.4f\n",floor(360));
}
```

14. ฟังก์ชัน **fabs(x)**

เป็นฟังก์ชันที่ใช้ในการหาค่าสัมบูรณ์ (absolute value) ของค่าคงที่ หรือตัวแปร X โดยที่ X เป็นค่าคงที่หรือตัวแปรที่เก็บค่าตัวเลข จุดทศนิยมที่มีค่าบวกหรือลบก็ได้

fabs(x);

```
/* math5.c */  
#include<stdio.h>  
#include<math.h>  
void main(void)  
{  
double x = -643.5678, y = 1645.7777;  
printf("Absolute value of  
x=%.5f\n",fabs(x));  
printf("Absolute value of  
y=%.5f\n",fabs(y));  
}
```

10.2 ฟังก์ชันที่เกี่ยวข้องกับตัวอักษร (character functions)

เป็นฟังก์ชันที่ใช้กับข้อมูลที่มีชนิดเป็น single char (ใช้เนื้อที่ 1 byte) เท่านั้น และก่อนที่จะใช้ฟังก์ชันประเภทนี้จะต้องใช้คำสั่ง `#include <ctype.h>` แทรกอยู่ตอนต้นของโปรแกรมจึงจะใช้ฟังก์ชันประเภทนี้ได้

ฟังก์ชันที่เกี่ยวข้องกับตัวอักษรที่ควรทราบ

`isalnum(ch)` `isalpha(ch)` `isdigit(ch)`

`islower(ch)` `isupper(ch)` `tolower(ch)`

`toupper(ch)` `isspace(ch)` `isxdigit(ch)`

1. ฟังก์ชัน `isalnum(ch)`;

เป็นฟังก์ชันที่ใช้ตรวจสอบว่าข้อมูลที่เก็บไว้ในตัวแปร `ch` เป็นตัวอักษรหรือตัวเลข (**letter or digit**) ถ้าข้อมูลที่เก็บไว้เป็นตัวอักษรหรือตัวเลขก็จะส่งค่ากลับที่เป็นจำนวนเต็มที่มีค่าไม่เท่ากับศูนย์มายังฟังก์ชัน และถ้าข้อมูลที่เก็บไว้ในตัวแปร `ch` ไม่ได้เก็บตัวอักษรหรือตัวเลขก็จะส่งค่ากลับที่มีค่าเป็นศูนย์มายังฟังก์ชัน

`isalnum(ch)`;

2. ฟังก์ชัน isalpha(ch);

เป็นฟังก์ชันที่ใช้ตรวจสอบว่าข้อมูลที่เก็บไว้ในตัวแปร ch เป็นตัวอักษร (letter) หรือไม่ ถ้าใช้ฟังก์ชันนี้จะให้ค่าส่งกลับเป็นตัวเลขจำนวนเต็มที่ไม่เท่ากับศูนย์ ถ้าไม่ใช่ฟังก์ชันนี้จะให้ค่าส่งกลับมาเป็นเลขศูนย์ (0)

isalpha(ch)

3. ฟังก์ชัน `isdigit(ch)`

เป็นฟังก์ชันที่ใช้ตรวจสอบว่าข้อมูลที่เก็บไว้ในตัวแปร `ch` เป็นตัวเลข 0-9 หรือไม่ ถ้าใช้ฟังก์ชันนี้จะให้ค่าส่งกลับเป็นเลขจำนวนเต็มที่ไม่เท่ากับศูนย์ ถ้าไม่ใช่ฟังก์ชันนี้จะไม่มีค่าส่งกลับ

```
isdigit(ch);
```

```
    /* char1.c */  
#include<stdio.h>  
#include<ctype.h>  
#include<conio.h>  
void main(void)  
{  
    char ch1 = 'A', ch2='8', ch3='+';  
    printf("%d\n",isalnum(ch1));  
    printf("%d\n",isalnum(ch2));  
    printf("%d\n",isalnum(ch3));  
    printf("%d\n",isalpha(ch1));  
    printf("%d\n",isalpha(ch2));  
    printf("%d\n",isalpha(ch3));  
}
```

4. ฟังก์ชัน `islower(ch)`

เป็นฟังก์ชันที่ใช้ตรวจสอบว่าตัวอักษรที่เก็บไว้ในตัวแปร `ch` เป็นตัวอักษรตัวเล็กหรือไม่ ถ้าใช้ฟังก์ชันนี้จะให้ค่าส่งกลับเป็นเลขจำนวนเต็มที่ไม่เท่ากับศูนย์ ถ้าไม่ใช่ฟังก์ชันนี้จะให้ค่าส่งกลับเป็นเลขศูนย์ (0)

`islower(ch)`

5. ฟังก์ชัน `isupper(ch)`

เป็นฟังก์ชันที่ใช้ตรวจสอบว่าตัวอักษรที่เก็บไว้ในตัวแปร `ch` เป็นตัวอักษรตัวใหญ่หรือไม่ ถ้าใช้ฟังก์ชันนี้จะให้ค่าส่งกลับเป็นเลขจำนวนเต็มที่ไม่เท่ากับศูนย์

`isupper(ch);`

6. ฟังก์ชัน `tolower(ch)`

เป็นฟังก์ชันที่ใช้เปลี่ยนตัวอักษรตัวใหญ่ที่เก็บไว้ในตัวแปร `ch` ให้เป็นตัวอักษรตัวเล็ก

```
tolower(ch);
```

7. ฟังก์ชัน `toupper(ch)`

เป็นฟังก์ชันที่ใช้เปลี่ยนตัวอักษรตัวเล็กที่เก็บไว้ในตัวแปร `ch` ให้เป็นตัวอักษรตัวใหญ่

```
toupper(ch);
```

```
/* char1.c */
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
void main(void)
{
    char ch1 = 'B', ch2='a';

    printf("%d\t%d\n",islower(ch1),islower(ch2));

    printf("%d\t%d\n",isupper(ch1),isupper(ch2));

    printf("%d\t%d\n",tolower(ch1),toupper(ch2));
}
```

3. ฟังก์ชันที่เกี่ยวกับสตริง (string functions)

เป็นฟังก์ชันที่ใช้เก็บข้อมูลชนิดสตริง (string) โดยก่อนใช้ฟังก์ชันประเภทนี้จะต้องใช้คำสั่ง `#include<string.h>` แทรกอยู่ตอนต้นของโปรแกรมเสียก่อน

`strlen(s)`

`strcmp(s1,s2)`

`strcpy(s)`

`strcat(s1,s2)`

4. ฟังก์ชันทั่วไปที่ใช้งานบ่อยๆ

1. ฟังก์ชัน `clrscr()` เป็นฟังก์ชันที่ใช้ในการลบข้อมูลออกจากจอภาพแบบ text mode

`clrscr(s);`

2. ฟังก์ชัน `gotoxy()` เป็นฟังก์ชันที่ตั้งให้ cursor เคลื่อนที่ไปยังตำแหน่งที่ระบุไว้ที่จอภาพ

x= ตำแหน่ง column บนจอภาพมีค่าตั้งแต่ 1 ถึง 79 ส่วน column ที่ 80 สงวนไว้

y= ตำแหน่ง row บนจอภาพมีค่าตั้งแต่ 1 ถึง 24 ส่วน row ที่ 25 สงวนไว้

`gotoxy() ;`

3. ฟังก์ชัน `creol()` เป็นฟังก์ชันที่ใช้ลบข้อความในบรรทัดที่ตัว `cursor` อยู่ โดยลบข้อความถัดจากตำแหน่งของ `cursor` ไปจนกระทั่งจบบรรทัด

`creol();`

4. ฟังก์ชัน `deline()` เป็นฟังก์ชันที่ลบข้อความทั้งบรรทัดที่มีตัว `cursor` จากนั้นก็เลื่อนข้อความในบรรทัดที่อยู่ข้างล่างมาแทนที่

`deline();`

5. ฟังก์ชัน `inpline()` เป็นฟังก์ชันที่ใช้แทรกบรรทัดว่าง 1 บรรทัด โดยแทรกอยู่ที่บรรทัดที่มี `cursor` อยู่

`inpline(s);`

6. ฟังก์ชัน `sizeof(x)` เป็นฟังก์ชันที่ใช้ตรวจสอบขนาดของตัวแปร `x` ว่ามีขนาดกี่ Byte

`sizeof(x);` หรือ `sizeof(type);`

`x` เป็นตัวแปรที่ต้องการตรวจสอบขนาด

`Type` เป็นชนิดตัวแปร เช่น `int, float, char, double` เป็นต้น

7. ฟังก์ชัน `system()` เป็นฟังก์ชันที่ช่วยให้สามารถเรียกใช้คำสั่งที่อยู่ใน MS-DOS มาใช้งานได้

`system("dos-command");`

8. ฟังก์ชัน `abort()` เป็นฟังก์ชันที่ใช้ยกเลิกการทำงานของโปรแกรมโดยทันทีทันที โดยไม่ว่าจะทำงานเสร็จหรือไม่ และไม่มีข้อความบอกว่า “Abnormal program termination” แสดงออกทางจอภาพด้วย

`abort(x);`

9. ฟังก์ชัน `abs(x)` เป็นฟังก์ชันที่ช่วยให้สามารถเรียกใช้คำสั่งที่อยู่ใน MS-DOS มาใช้งานได้

`abs(x);`

ตัวอย่างเช่น `int x = -65`

`printf(“%d”, abs(x));`

ผลลัพธ์ 65

10. ฟังก์ชัน `labs(x)` เป็นฟังก์ชันที่ใช้หาค่าสัมบูรณ์ของ `x` โดยที่ `x` เป็นตัวแปรที่เก็บตัวเลข

`labs(x);`

11. ฟังก์ชัน `atoi(s)` เป็นฟังก์ชันที่เปลี่ยนค่า `string` ให้เป็นตัวเลขจำนวนเต็ม (`integer`) ที่สามารถไปคำนวณได้

`atoi(s);`

ตัวอย่างเช่น `char numstr[20]= "1000"; int d;`

`d=atoi(numstr);`

`printf("%d", d);`

ผลลัพธ์ 1000

12. ฟังก์ชัน `atof(s)` เป็นฟังก์ชันที่ใช้เปลี่ยนค่า `string` ให้เป็นตัวเลขจำนวนทศนิยมที่สามารถนำไปคำนวณได้

`atof(s);`

13. ฟังก์ชัน `atol(s)` เป็นฟังก์ชันที่เปลี่ยนค่า `string` ให้เป็นตัวเลขจำนวนเต็มชนิด `long integer` ที่สามารถนำไปคำนวณได้

`atol(s);`