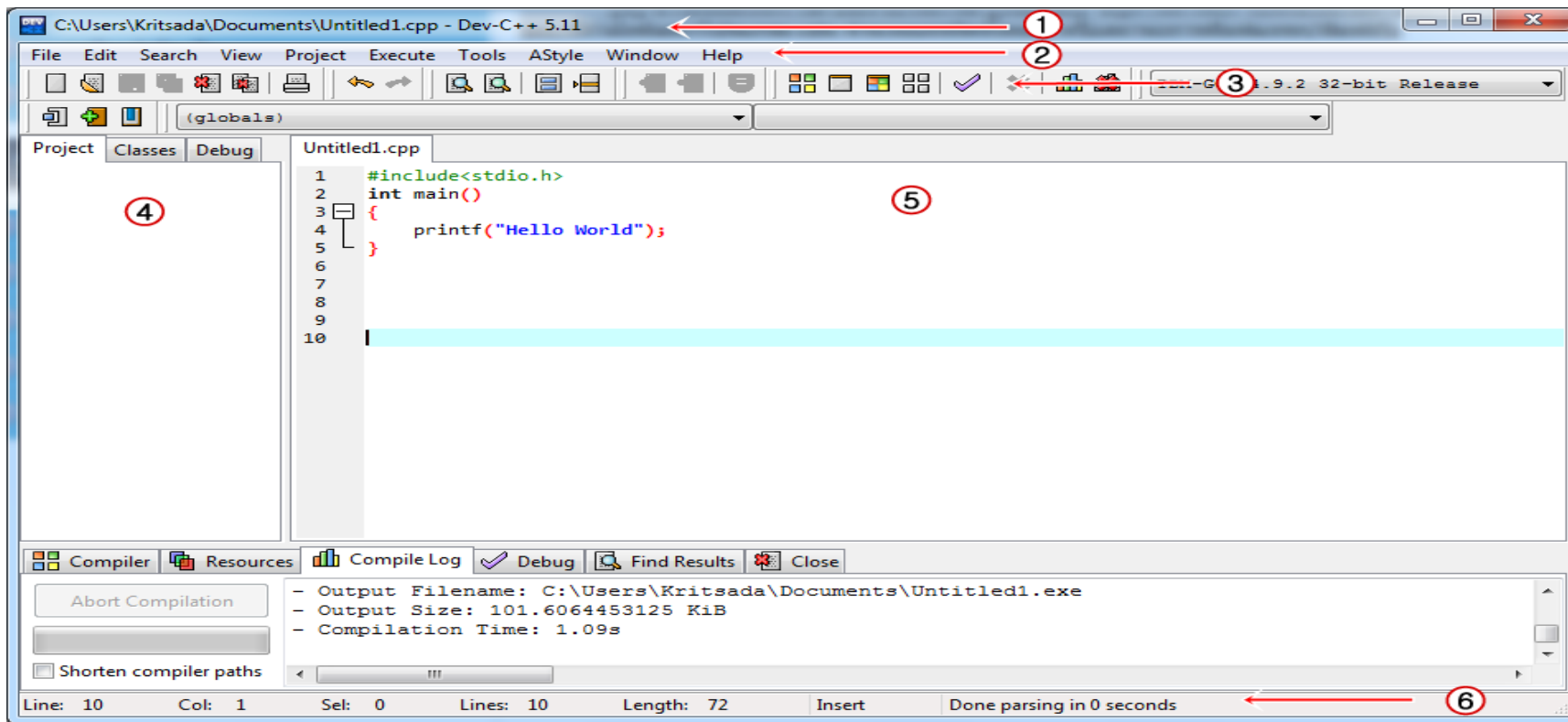




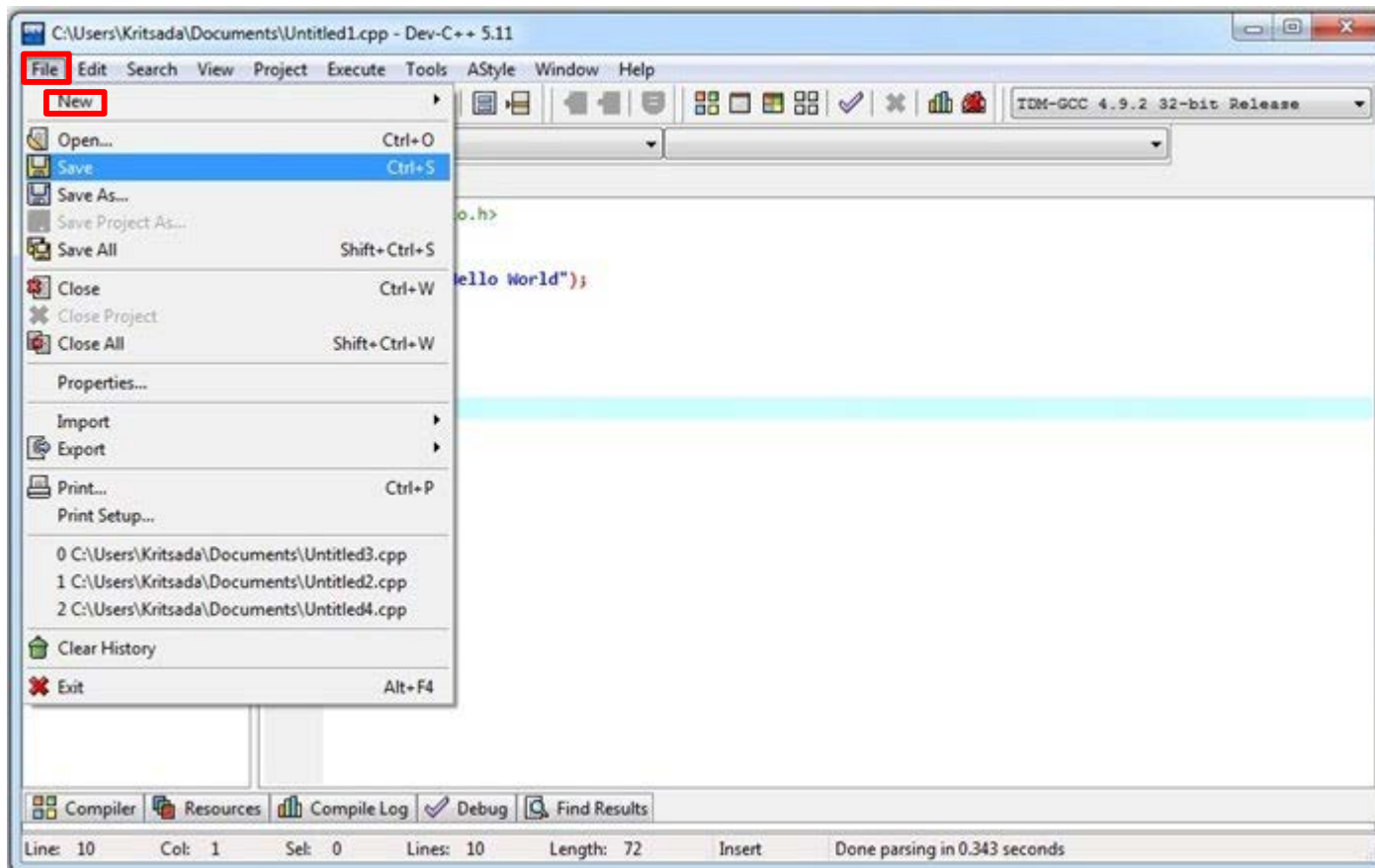
การใช้งาน Dev c++ เบื้องต้น



- (1) ส่วนของไตเติลบาร์ (Title Bar) เป็นแถบที่อยู่บนสุดของโปรแกรม มีสีน้ำเงิน และจะมีชื่อของชุดพัฒนาโปรแกรมภาษาซี คือ Dev-C++
- (2) ส่วนของเมนูบาร์ (Menu Bar) ส่วนนี้เป็นส่วนของเมนูคำสั่งต่าง ๆ
- (3) ส่วนของทูลบาร์ (Tool Bars) เป็นส่วนของเครื่องมือ ที่ช่วยอำนวยความสะดวกในการเขียนและพัฒนาโปรแกรม
- (4) ส่วนของการแสดง Project/Classes/Debug เป็นส่วนที่อยู่ทางด้านซ้ายของโปรแกรม ใช้เพื่อแสดง Project หรือ Class ต่าง ๆ ของโปรแกรม
- (5) ส่วนของพื้นที่การเขียนโปรแกรม (Editor) เป็นส่วนของพื้นที่ทำงานในการเขียนโค้ดโปรแกรมภาษา C
- (6) ส่วนแสดงสถานะของโปรแกรม (Status) อยู่ด้านล่างสุด ใช้เพื่อบอกสถานะต่าง ๆ ในขณะที่กำลังพัฒนาโปรแกรม

# 1. การสร้างไฟล์ใหม่

สร้างไฟล์ใหม่ผ่านเมนู File -> New ->Source File หรือใช้คีย์ลัดกดปุ่ม Ctrl+N



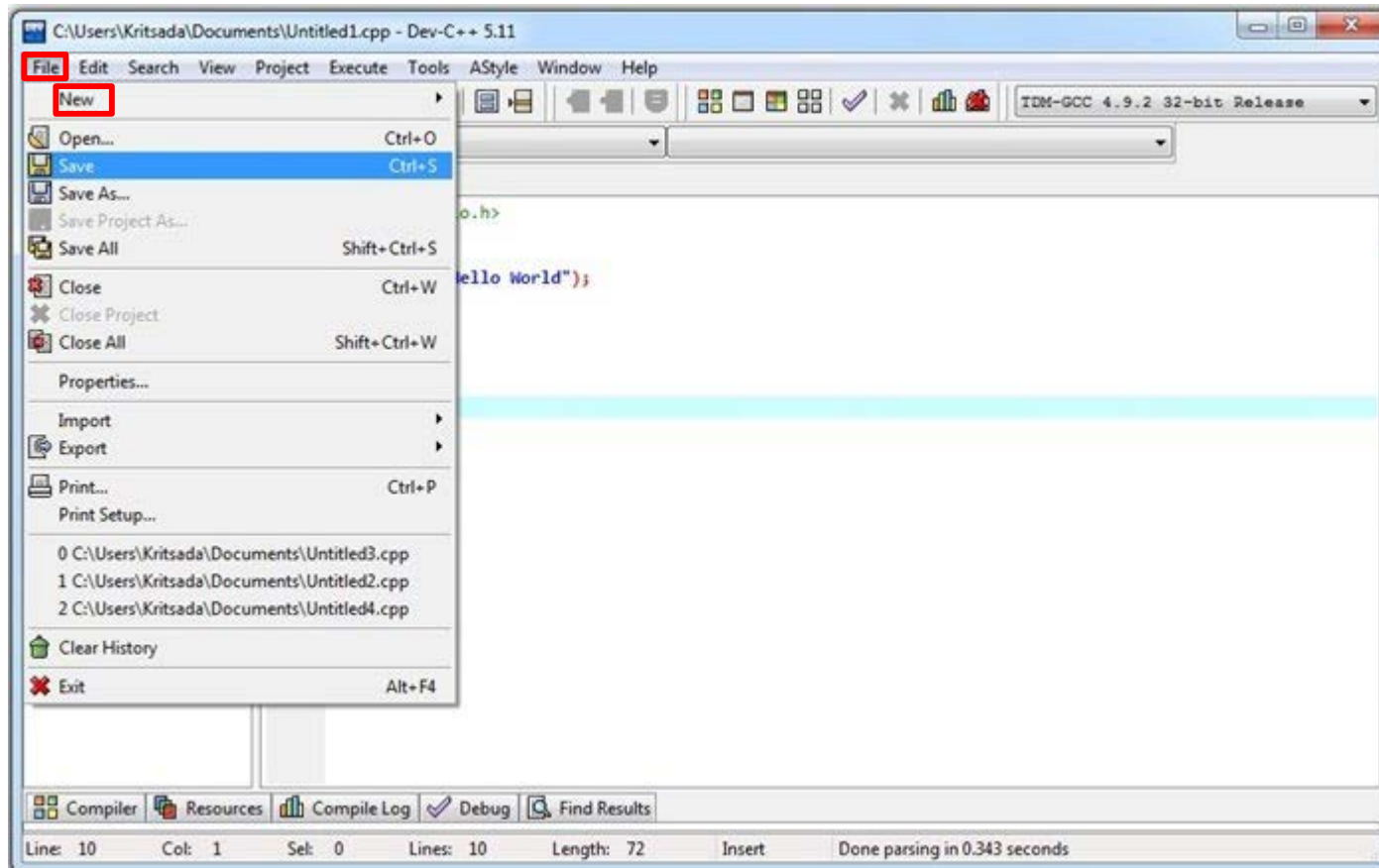
The screenshot shows the Dev-C++ IDE interface. The main window displays a C++ program in a file named 'Untitled1.cpp'. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     printf("Hello World");
5 }
6
7
8
9
10 |
```

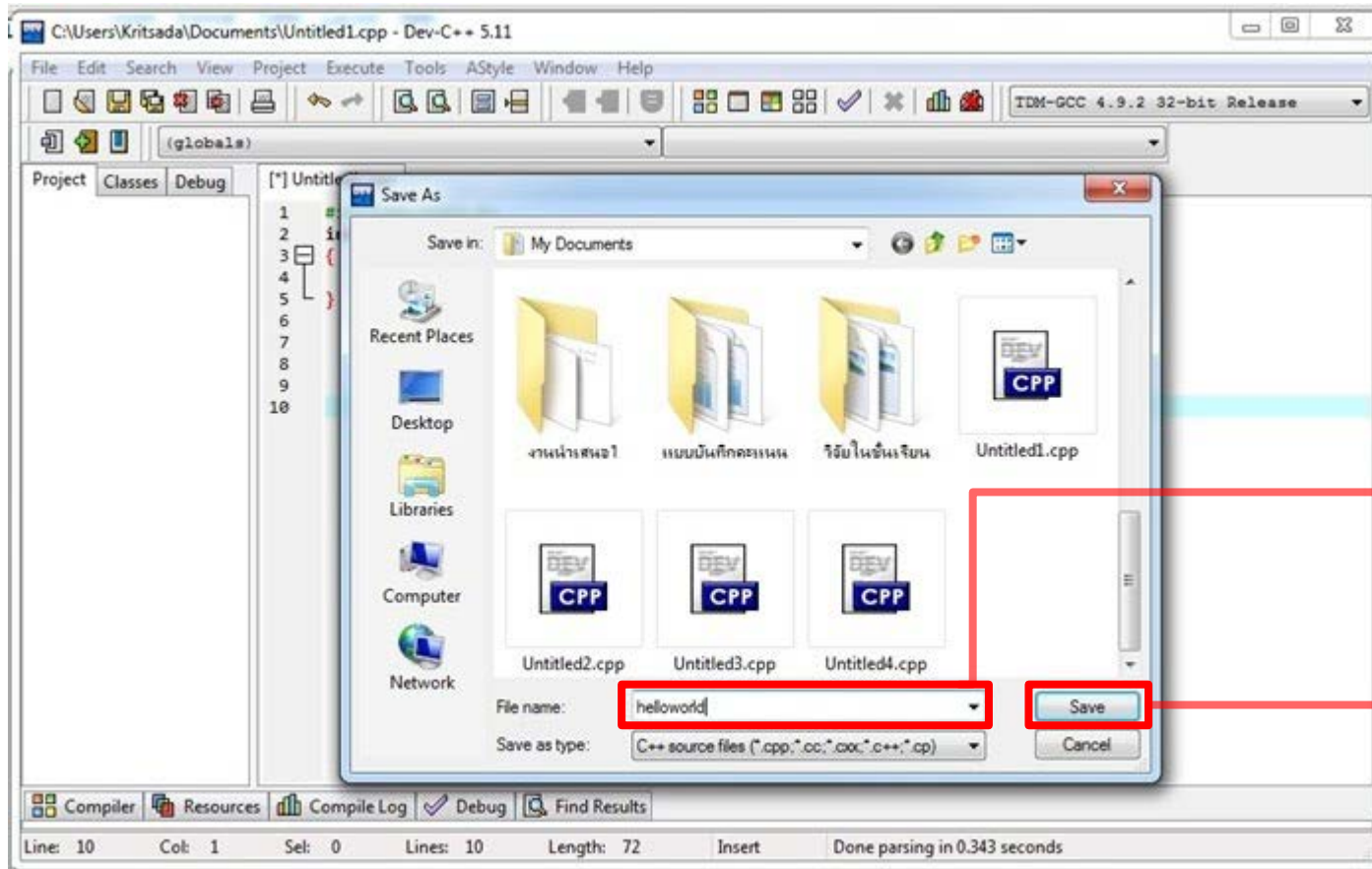
A red rectangular box highlights the entire code area from line 1 to line 10. A light blue horizontal highlight is present on line 10. The status bar at the bottom indicates 'Line: 10 Col: 1 Sel: 0 Lines: 10 Length: 72 Insert Done parsing in 0.343 seconds'.

พื้นที่สำหรับการเขียนโค้ด

การบันทึก (Save) โดยไปคลิกที่เมนู File --> Save หรือกดปุ่มคีย์ลัด Ctrl + S



# การตั้งชื่อเพื่อบันทึกชื่อไฟล์



ชื่อไฟล์

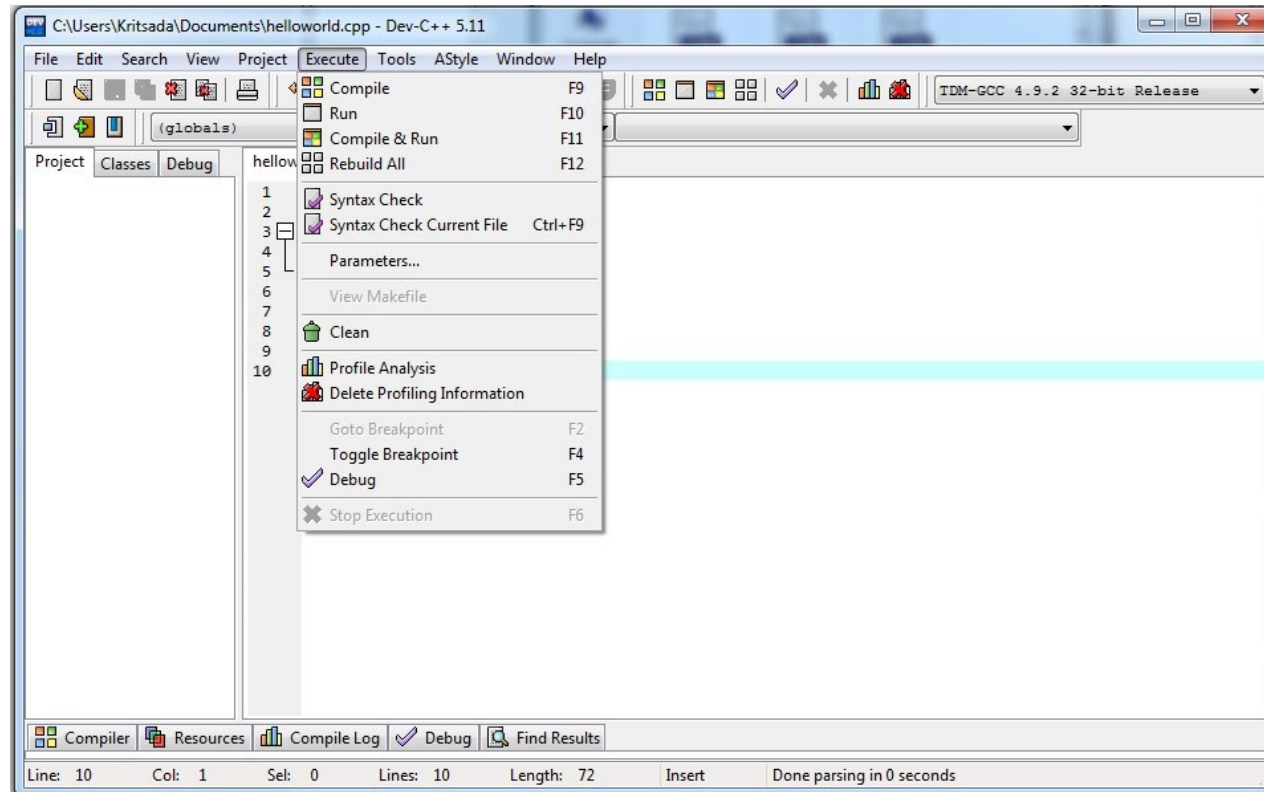
ปุ่มบันทึก

## 2. การ Compile (คอมไพล์) โปรแกรม

การ Compile คือ การตรวจสอบ Source Code ที่เขียนว่าถูกต้องตามโครงสร้างของภาษาหรือไม่

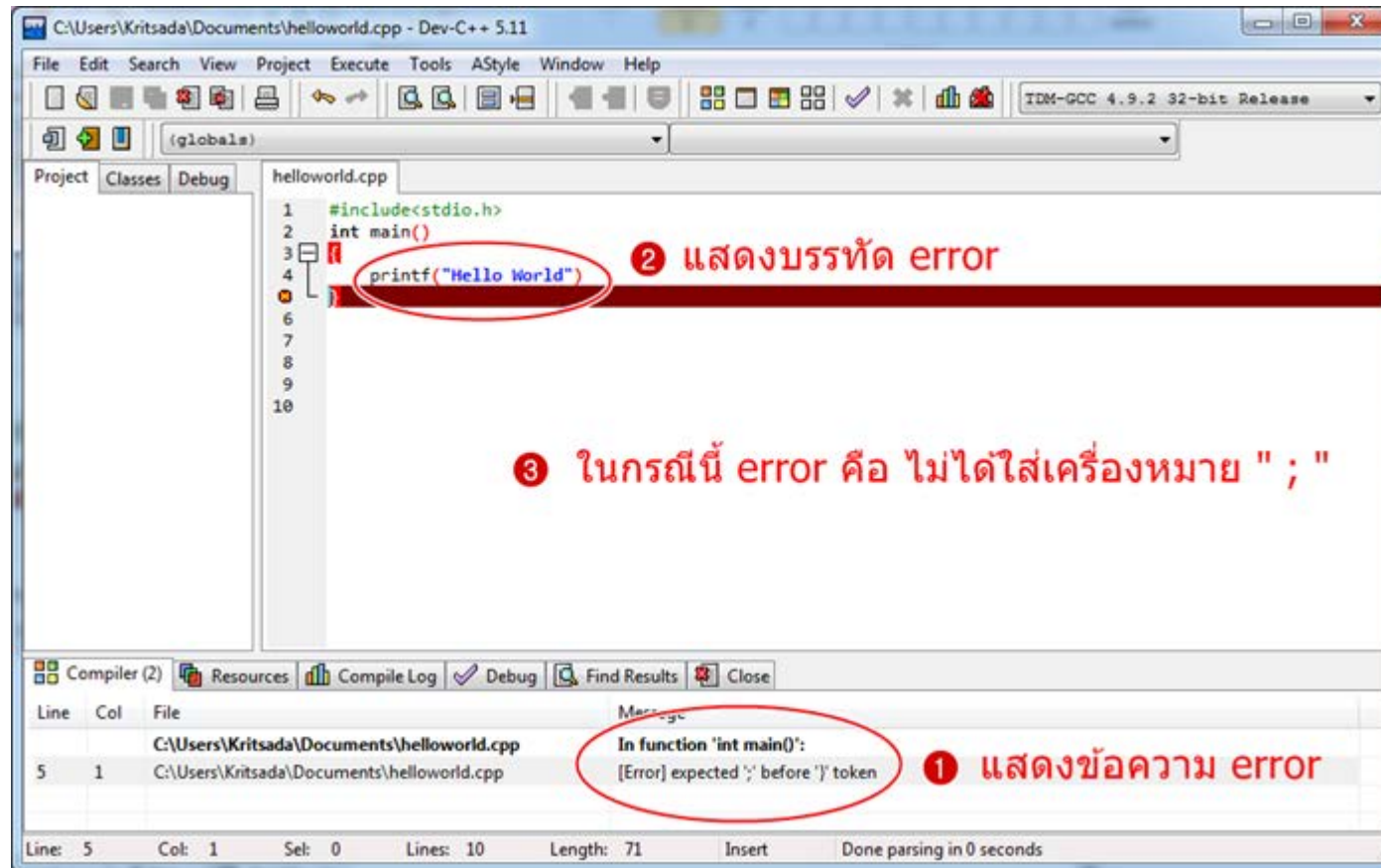
---

วิธี Compile คือ Execute --> Compile หรือ กดคีย์ลัดที่เป็นพิมพ์ F9





หากมีการ error ของโปรแกรมสามารถอ่านสาเหตุของการ error ได้จากหมายเลข 1

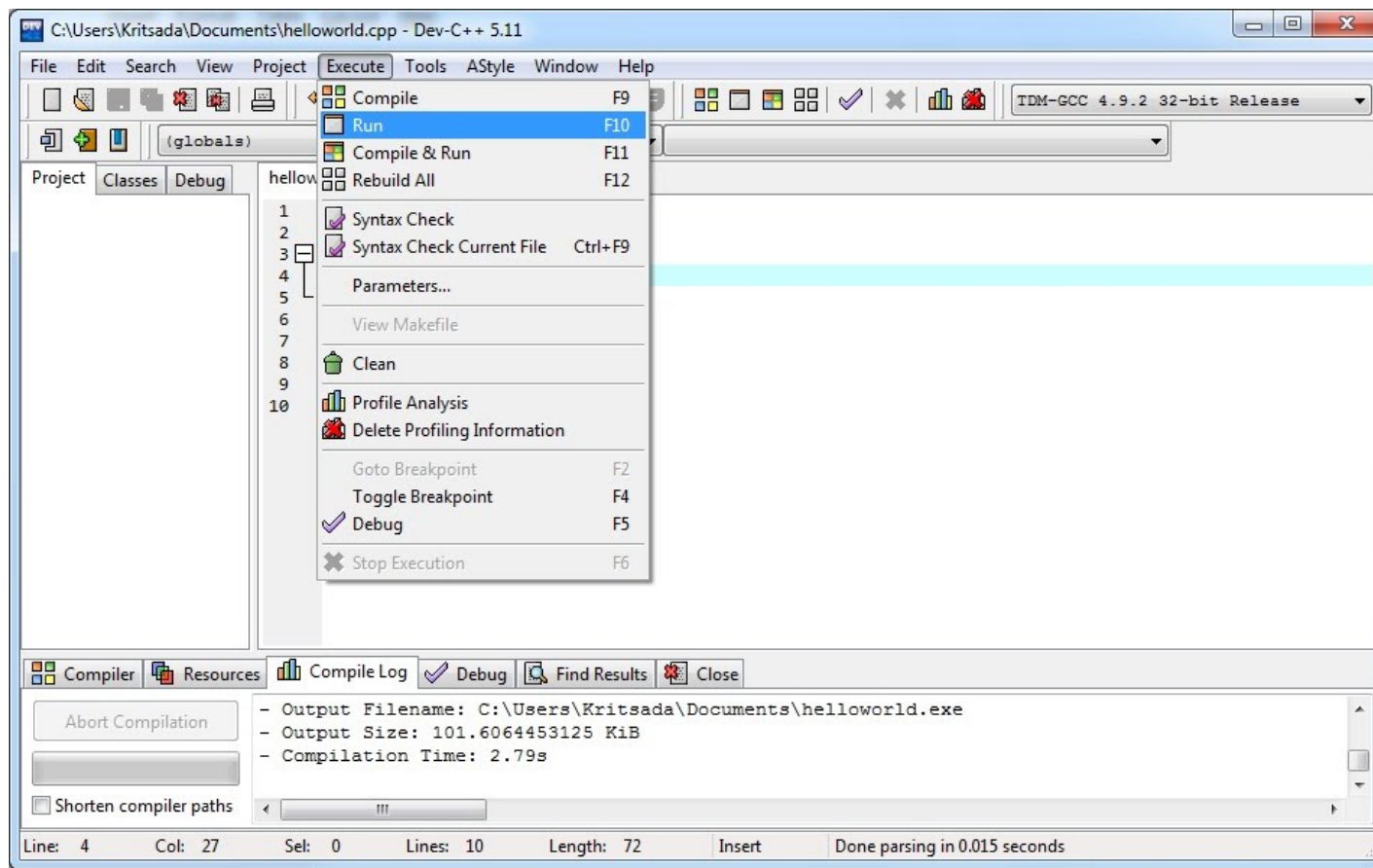




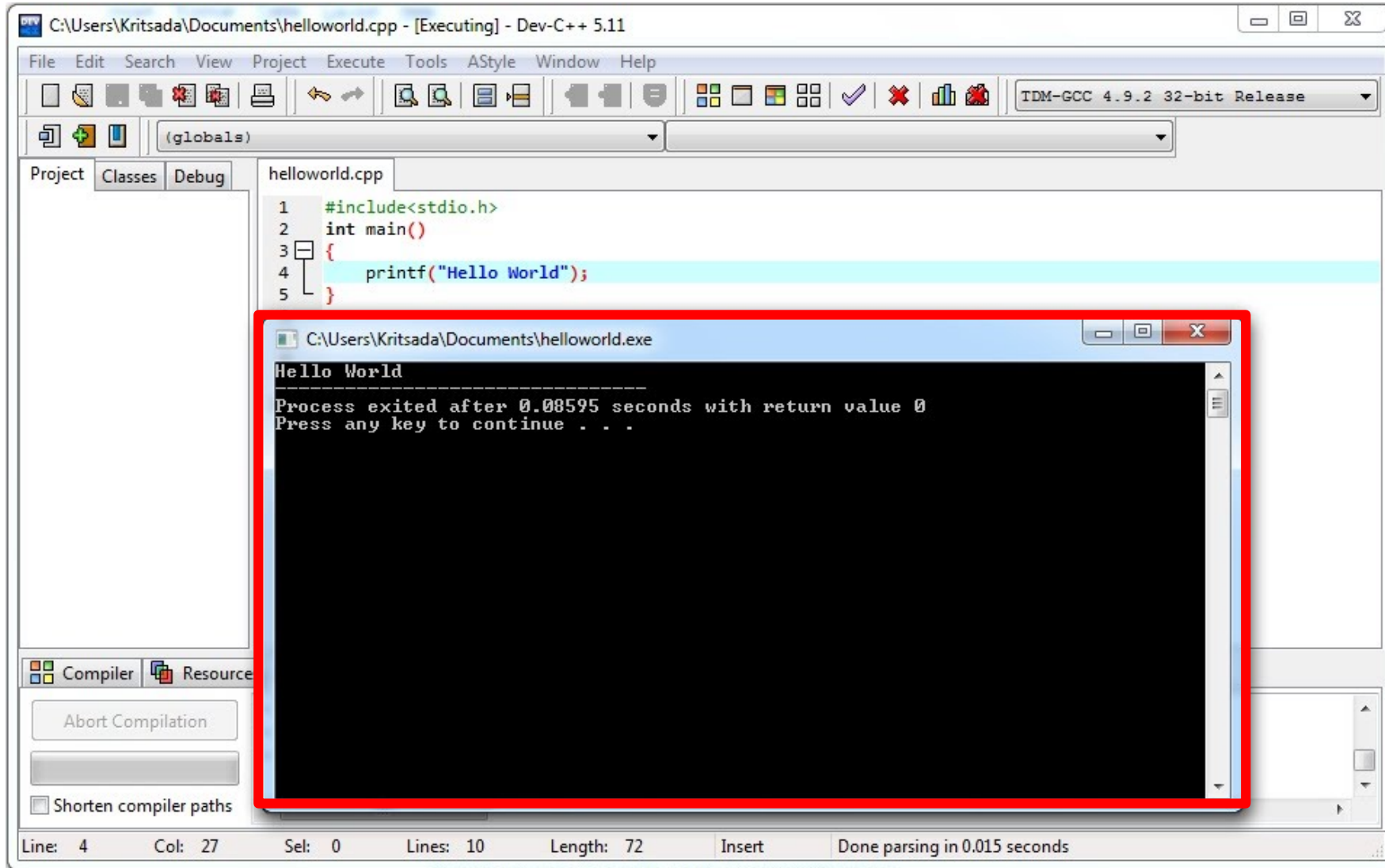
### 3. การเรียกโปรแกรมทำงาน (รันโปรแกรม)

การรันโปรแกรม คือ การเรียกโปรแกรมที่เราเขียนขึ้นมาทำงาน

วิธีการ Run คือคลิกที่เมนู Execute --> Run หรือคีย์ลัด F10



# ภาพแสดงตัวอย่างผลการ Run โปรแกรม ซึ่งจะปรากฏให้เห็นบน Dos Mode



# 4. ตัวอย่างการใช้งานโปรแกรม

## การเขียนโปรแกรมรับค่าและแสดงผลเบื้องต้น

```
#include <stdio.h>

int main() {
    int x ;
    printf( _Format: "Hello, World!\n");
    scanf( _Format: "%d" ,&x) ;
    printf( _Format: "Anser is = %d" ,x) ;
    return 0;
}
```

ใช้บอกกับคอมพิวเตอร์ให้นำเฮดเดอร์ไฟล์ที่ระบุคือ stdio.h  
เข้าร่วมในการแปลโปรแกรม

เป็นการกำหนดฟังก์ชันหลัก(main) ให้อยู่ในรูปจำนวนเต็ม  
(integers) หรือ int ในโปรแกรม

เป็นการกำหนดตัวแปรเพื่อเก็บข้อมูล ในที่นี้ ให้ตัวแปร x  
เป็นรูปแบบของจำนวนเต็ม(integers) หรือ int ในโปรแกรม

เป็นการแสดงผล โดยใช้คำสั่ง printf เพื่อแสดงข้อมูลออก  
ทางหน้าจอ

เป็นการรับค่าข้อมูล โดยใช้คำสั่ง scanf เป็นคำสั่งเพื่อรับ  
ข้อมูลจากแป้นพิมพ์

# 5. ชนิดของข้อมูลและตัวดำเนินการ

เขียนโปรแกรมโดยใช้ข้อมูลและตัวดำเนินการ

```
#include <stdio.h>

int main(){
int x,y,sum,minus,mul ;
float div ;

printf( _Format: "enterNumber_1:");
scanf( _Format: "%d" ,&x);
printf( _Format: "enterNumber_2:");
scanf( _Format: "%d" , &y) ;

sum=x+y ;
minus = x-y ;
mul = x*y ;
div = x/y ;

printf( _Format: "Sum: %d\n",sum);
printf( _Format: "minus: %d\n",minus);
printf( _Format: "mul: %d\n" , mul);
printf( _Format: "div: %.2f",div) ;

}
```

การกำหนดตัวแปร... int = จำนวนเต็ม , float = ทศนิยม

ส่วนของการรับค่าและเก็บค่าไปในตัวแปร

ส่วนของการใช้ตัวดำเนินการด้านการคำนวณเบื้องต้น

ส่วนของการแสดงผลลัพธ์

ผลลัพธ์

```
enterNumber_1:10
enterNumber_2:2
Sum: 12
minus: 8
mul: 20
div: 5.00
```

## 6.Control Structure : Selection

เพื่อให้เข้าใจโครงสร้างการทำงานของโปรแกรมเลือกทิศทางการทำงานได้ โดยใช้ if ,if-else ,if –else if

### คำสั่ง if

การเลือกทำโดยใช้คำสั่ง if โปรแกรมจะทำงานทางเดียว

```
if (condition) statement ;
```

- condition : เงื่อนไขที่กำหนดขึ้น เพื่อใช้พิจารณาว่าจะทำหรือไม่ทำตามคำสั่ง โดยเงื่อนไขอาจจะอยู่ในรูปของนิพจน์การคำนวณ และเปรียบเทียบ หรือเป็นค่าของตัวแปรก็ได้และจะต้องเขียนไว้ภายในเครื่องหมาย()

- statement : คำสั่งที่จะให้ทำงานถ้าผลการตรวจสอบเงื่อนไขออกมาเป็นจริง (True)

### การใช้คำสั่ง if

```
if(condition)
```

```
{
```

```
    statement-1 ;
```

```
    statement-2 ;
```

```
    statement-3 ;
```

```
    ...
```

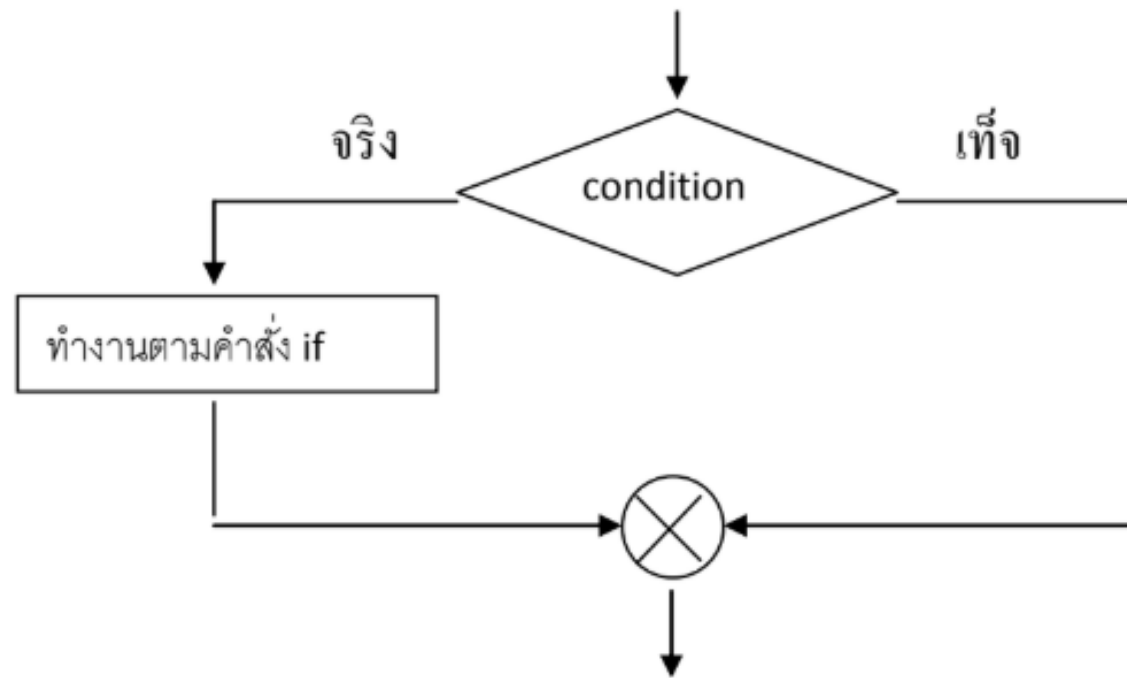
```
    statement-n ;
```

```
}
```

## ลักษณะการทำงานของคำสั่ง if

### แผนภาพจำลองการทำงานของคำสั่ง if

แผนภาพจำลองการทำงานของคำสั่ง if แสดงดังรูปต่อไปนี้



## ตัวอย่างการเขียนโปรแกรมโดยใช้คำสั่ง if

```
#include <stdio.h>
int point ;
void main()
{
    printf( _Format: "Enterexamination points :");
    scanf( _Format: "%d",&point);

    if(point >= 50)
    {
        printf( _Format: "You passed , congratulation");
    }
}
```

ผลลัพธ์

```
Enterexamination points : 50
You passed , congratulation
```



## คำสั่ง if-else

การเลือกทำโดยใช้คำสั่ง if-else จะใช้เมื่อต้องการทำงานแบบทางเลือก 2 ทาง

```
if(condition)
```

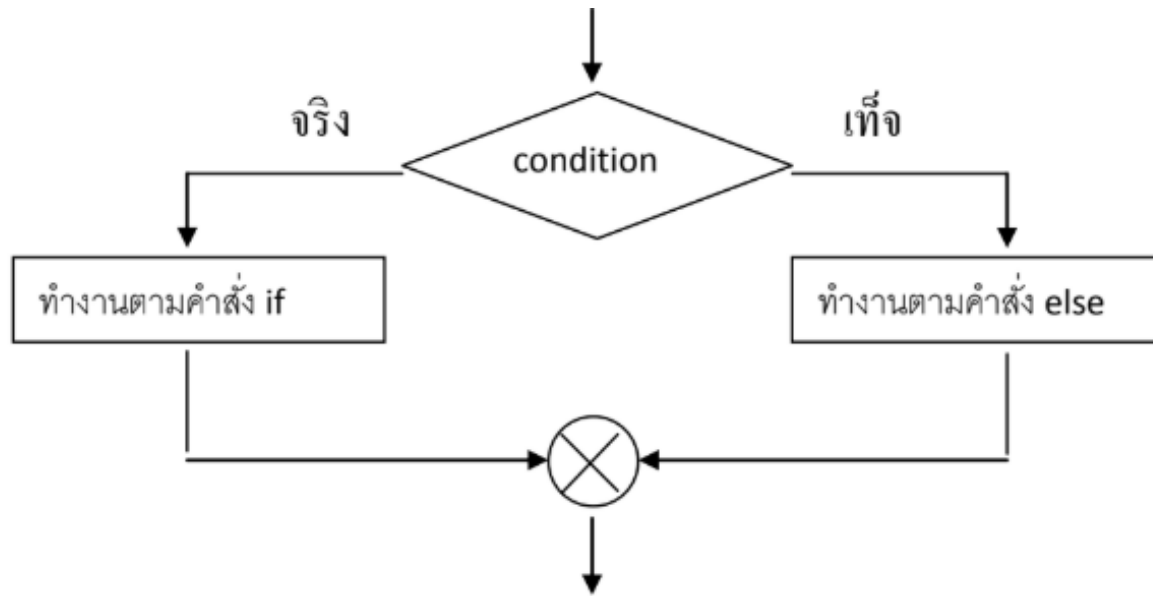
```
    statement ;
```

```
else
```

```
    statement ;
```

## ลักษณะการทำงานของคำสั่ง if-else

### แผนภาพจำลองการทำงานของคำสั่ง if-else



## ตัวอย่างการใช้คำสั่ง if-else

```
#include <stdio.h>
int point ;
void main()
{
    printf( _Format: "Enter your examination points : ");
    scanf( _Format: "%d",&point);
    if(point >= 50)
        printf( _Format: "You passed ,Congratulation.\n");
    else
        printf( _Format: "Sorry , you didn't pass.\n");
}
```

ผลลัพธ์

```
Enter your examination points :60
You passed ,Congratulation.
```

```
Enter your examination points :20
Sorry , you didn't pass.
```

## คำสั่ง if-else if

การเลือกทำโดยใช้คำสั่ง if-else if จะใช้เมื่อต้องการทำงานแบบทางเลือก 2 ทาง โดยแต่ละทางเลือกจะมีเงื่อนไขต่างกัน

ลักษณะการทำงานของคำสั่ง if-else if

```
if(condition-1)
    statement ;
else if (condition-2)
    statement ;
else if(condition-3)
    statement ;
...
else if(condition-n)
    statement ;
else
    statement ;
```

## ตัวอย่างการใช้คำสั่ง if-else if

```
1  #include <stdio.h>
2  int main(void)
3  {
4  int score;
5  printf("Input score: ") ;
6  scanf("%d",&score);
7  if ((score >= 80) && (score <= 100)) printf("A");
8  else if ((score >= 70) && (score < 80)) printf("B");
9  else if ((score >= 60) && (score < 70)) printf("C");
10 else if ((score >= 50) && (score < 60)) printf("D");
11 else printf("F");
12 return 0;
13 }
```

ผลลัพธ์

```
Input score: 72
```

```
B
```

```
-----
Process exited after 8.072 seconds with return value 0
Press any key to continue . . .
```

## 7. มารู้จักกับ loop

**for loop** หลักการคือ กำหนดค่า ตัวนับค่าให้มันก่อน สมมุติ คือ ตัวแปร  $i$  ให้เป็น 0 จากนั้นให้มันทำไปทีละรอบ ก็เพิ่มค่าทีละ 1 โดยใช้  $i++$  แต่ละรอบก็ตรวจสอบว่ามันครบรึยัง โดยเงื่อนไข  $i < 5$  หรือไม่ ถ้ามันครบ 5 รอบ มันก็จะออกจาก for loop นั้นเอง

```
“  
for( กำหนดค่าเริ่มต้น ; เงื่อนไข ; เปลี่ยนแปลงค่า ){  
    // do something  
}  
”
```

## ลักษณะการเขียน for loop

```
#include <stdio.h>
int main() {
    int i;
    for(i = 0 ; i < 5;i++){
        printf(_Format: "\n Hello world");
    }
}
```

ผลลัพธ์

```
Hello world
Hello world
Hello world
Hello world
Hello world
```



**While loop** คือ ทำวนรอบแต่ไม่รู้จำนวนรอบที่แน่นอน อาจทำเพื่อคำนวณอะไรบางอย่าง

```
“while (เงื่อนไข){  
    }  
”
```

**หมายเหตุ** ถ้าเงื่อนไขเป็นจริงก็จะทำใน { } ไปเรื่อยๆ ดังนั้น อย่าลืมเปลี่ยนแปลงค่าในการเช็คเงื่อนไขด้วย จะทำจนกว่าเงื่อนไขเป็นเท็จ

## ลักษณะการเขียน while loop

```
#include <stdio.h>
main(){
    int data = 2;

    while(data % 19 != 0){
        data += 3;
    }

    printf(_Format: "data = %d", data);
}
```

ผลลัพธ์

```
data = 38
```

## 8. มารู้จักกับ Array

ถ้าผมต้องการเก็บข้อมูลคะแนนนักเรียน 6 คน วิธีการแบบดิบๆ คือ ผมสามารถประกาศตัวแปร `int score 1, score2 , ..` ครบ 6 ตัว

ซึ่งมันก็ดูลำบาก มีหลายตัวแปร ตั้งหลายชื่อ

**array** คือ ตัวแปรชุด เอาไว้เก็บข้อมูลเป็นชุด โดยมีตัวแปรชื่อเดียว

สัญลักษณ์ คือ `[]`

ประกาศตัวแปร `int` เป็นชุด พร้อมกำหนดค่า คะแนนนักเรียน

```
int score[] = { 70 , 82 , 63 , 89 , 98 , 100 };
```

หรือทำเป็นตัวแปร char เก็บเกรดของวิชา

```
char grade[] = { 'A' , 'B' , 'C' , 'D' , 'F' };
```

การเข้าถึงข้อมูลใน array จะใช้ตัวเลขลำดับ เรียกว่า index

วิธีการ คือ ชื่อตัวแปร ตามด้วย [ ] ข้างใน [ ] คือ เลข index โดยจะเริ่มตั้งแต่ 0 ถึง n-1

ถ้า มี 6 ค่า ก็จะมี index 0 1 2... 5

เช่น ถ้าจะแสดงค่าคะแนนของนักเรียนคนที่ 1 ซึ่งก็จะมี index = 0

```
printf("%d",score[0]);
```

ถ้าจะเปลี่ยนค่า score[0] ก็ทำได้เหมือนตัวแปรทั่วไป

```
score[0] = 99;
```

ตัวอย่างการแสดงผลของทั้ง 6 คน โดยค่า index = 0 1 2 ... 5

```
#include <stdio.h>
main(){
    int score[] = { 70 , 82 , 63 , 89 , 98 , 100 };
    printf(_Format: "\n %d", score[0]);
    printf(_Format: "\n %d", score[1]);
    printf(_Format: "\n %d", score[2]);
    printf(_Format: "\n %d", score[3]);
    printf(_Format: "\n %d", score[4]);
    printf(_Format: "\n %d", score[5]);
}
```

ผลลัพธ์

```
70
82
63
89
98
100
```