

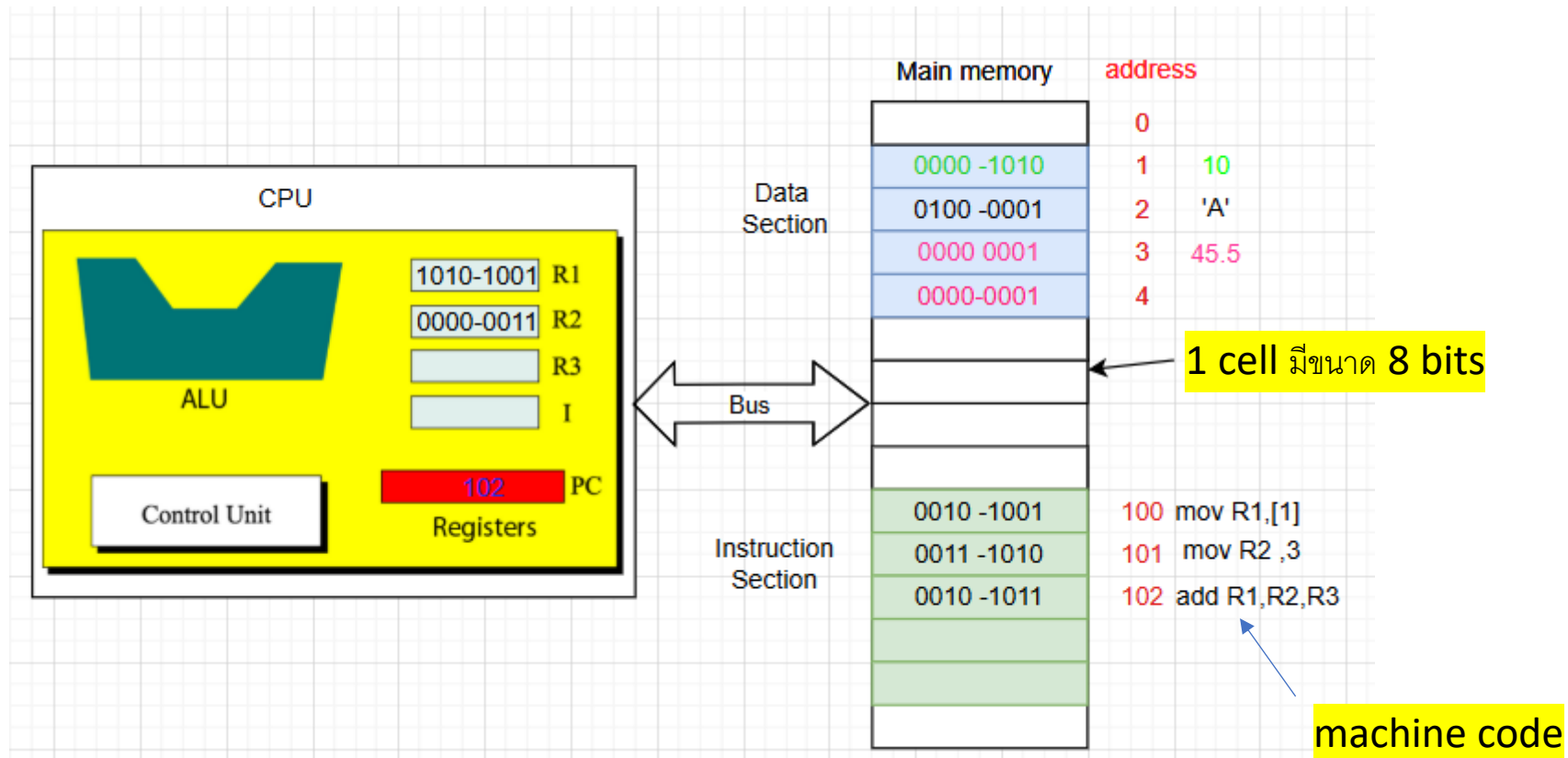
# Variable

## แนวคิดของตัวแปร

# หัวข้อ

- Von Neumann architecture
- หลักการของตัวแปร (**variable**) บนภาษาโปรแกรม

# สถาปัตยกรรม Von Neumann

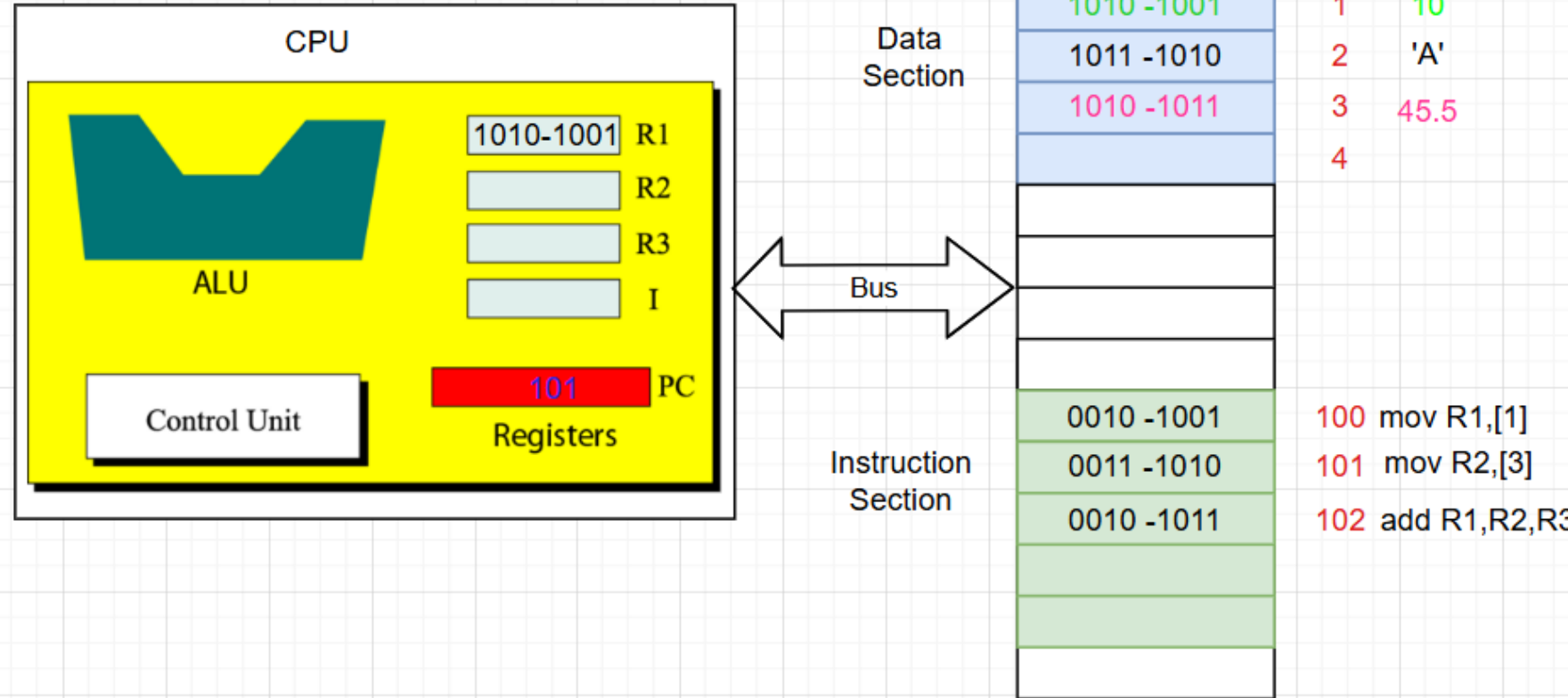


หลักการ

1. โปรแกรมและข้อมูลอยู่ในหน่วยความจำเดียวกัน ขณะโปรแกรมประมวลผล
2. CPU execute คำสั่งทีละคำสั่งเรียงต่อเนื่องกัน
3. ข้อมูล (data) และ คำสั่งทุกอย่างถูกเข้ารหัสด้วยระบบเลขฐาน 2 (binary system)

# การแทนรหัสคำสั่ง

- คำสั่ง (instruction) ที่ใช้  
การทำงานของ **cpu** ถูกเข้ารหัส  
ด้วยระบบเลขฐาน 2  
(binary system) ซึ่ง  
**CPU** แต่ยี่ห้อ ก็จะมีรูปแบบ  
คำสั่งที่ไม่เหมือนกัน

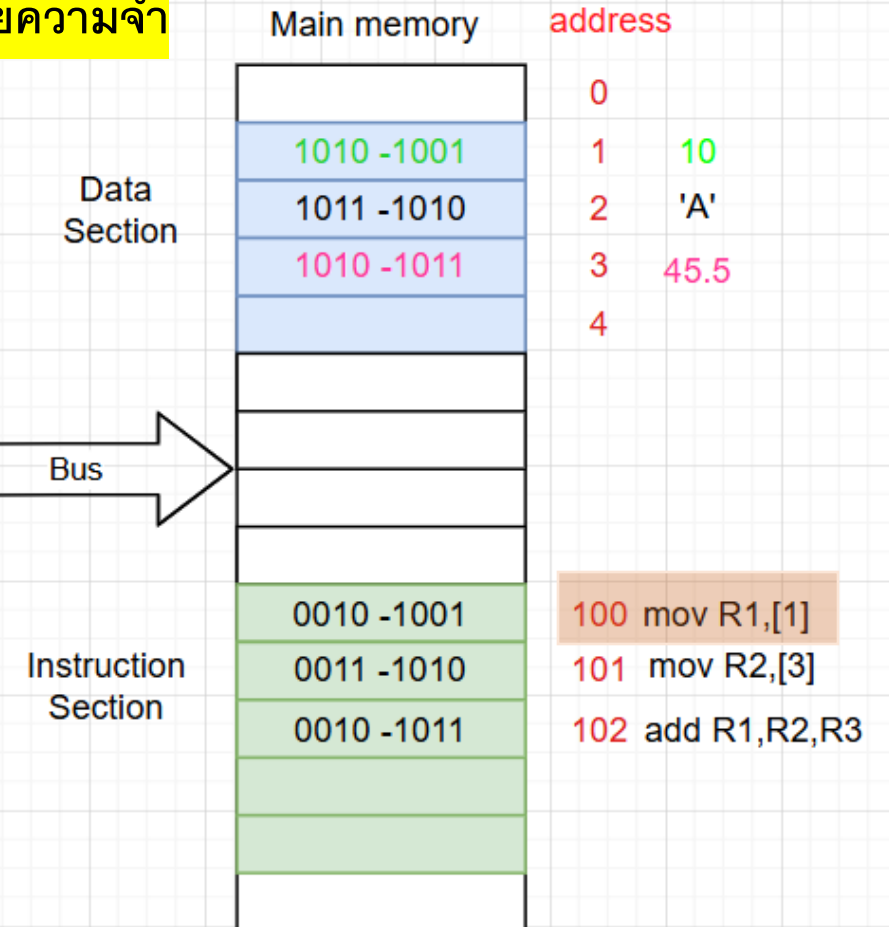
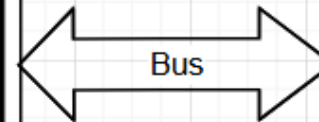
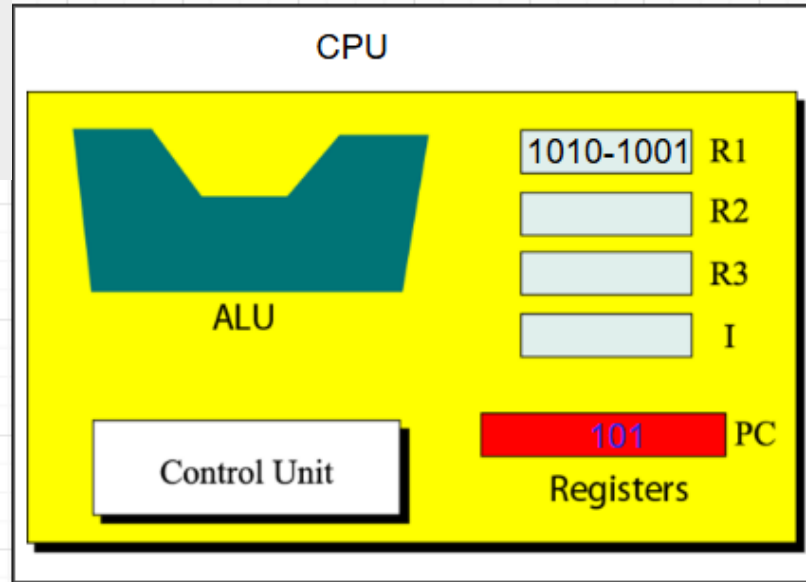


# การจัดเก็บข้อมูลและการเข้าถึงข้อมูลในหน่วยความจำ (1/2)

- การเข้าถึงหน่วยความจำ (**memory access**)
  - ขณะที่โปรแกรมประมวลผล เมื่อโปรแกรมต้องการอ่านหรือเขียนข้อมูลที่อยู่บนหน่วยความจำ โปรแกรมจะต้องมีการระบุตำแหน่ง (**address**) ของหน่วยความจำ

100: Mov R1, [1]

- การอ่านข้อมูลที่ addr 1 มาจัดเก็บใน R1



# การจัดเก็บข้อมูลและการเข้าถึงข้อมูลในหน่วยความจำ (2/2)

- การจัดสรร **cell** (allocate cell in memory)
  - โปรแกรมต้องมีจอง (**allocate**) จำนวน **cell** ให้เพียงพอสำหรับเก็บข้อมูล

Main memory			address		
			0		
	0000 -1010		1	10	
	0100 -0001		2	'A'	
	0000 0001		3	257	
	0000-0001		4		

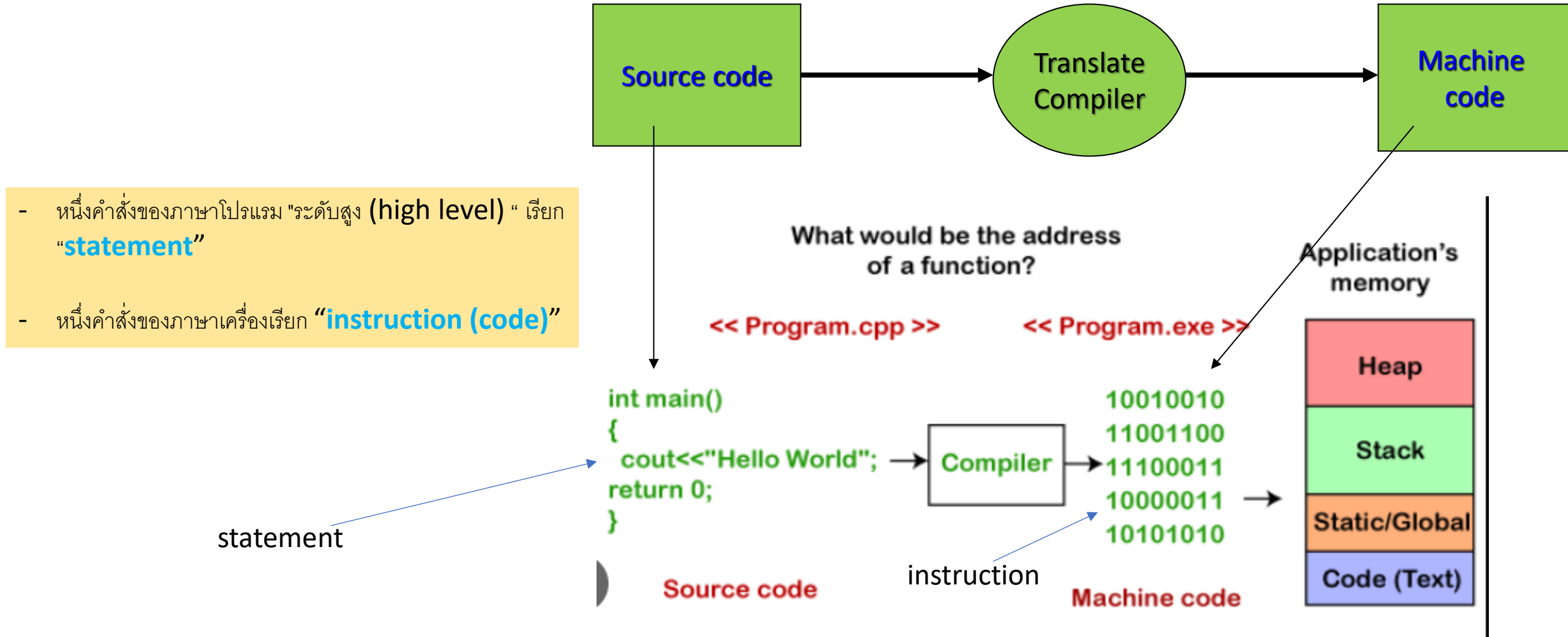
ธ.ค.	Hex	ไบนารี	HTML	ถ่าน	คำอธิบาย
65	41	01000001	&#65;	A	Capital A
66	42	01000010	&#66;	B	Capital B

dec	binary
10	0000-1010
257	1-0000-0001

# ปัญหาในการพัฒนาโปรแกรม

- คำสั่งอยู่รูปแบบรหัสของ “เลขฐาน 2”
- ข้อมูลที่นำมาประมวลในระบบ ต้องถูกเข้ารหัส (**encode**) เป็นเลขฐาน 2
- การจัดเก็บข้อมูลในหน่วยความจำต้องมีการกำหนด “จำนวน **cell** ของ **memory**” เพื่อให้เพียงพอต่อการจัดเก็บข้อมูล
- การจัดเก็บข้อมูลในหน่วยความจำต้องมีการ “ระบุตำแหน่ง **address** ของ **memory**” เพื่อจัดเก็บข้อมูล เพราะทุกครั้ง que **cpu** อ้างถึงข้อมูลจะต้องมีการระบุตำแหน่งของหน่วยความจำ

# ภาษาโปรแกรม (Programming Languages)





# Statement vs Instruction

- **หนึ่ง Statement** อาจประกอบด้วย หลาย **statement** ย่อยขึ้นอยู่โครงสร้างภาษาโปรแกรม

```
x := x + 1;  
if (x > 0)  
begin  
    writeln("จำนวนเป็นบวก");  
    x := x + 2;  
end;
```

Statement : หนึ่งคำสั่งในภาษาระดับสูง

Instruction : หนึ่งคำสั่งในภาษา machine code

```
MOV AX, x    ; instruction  
ADD AX, 1    ; instruction  
MOV x, AX    ; instruction
```

```
CMP x, 0      ; เปรียบเทียบ x กับ 0  
JLE skip_print ; ถ้า x <= 0 ให้ข้าม  
CALL writeln  ; เรียกฟังก์ชันพิมพ์  
MOV AX, x     ; instruction  
ADD AX, 2     ; instruction  
MOV x, AX     ; instruction
```

skip\_print:

# ตัวแปร (Variable)

- **ตัวแปร (บนภาษาโปรแกรม)** คือ สิ่งที่ใช้แทนตำแหน่งและขนาดของหน่วยความจำที่ถูกจัดสรร เพื่อเก็บข้อมูลขณะที่โปรแกรมประมวลผล โดยสามารถ**เปลี่ยนแปลงค่าได้**ระหว่างการทำงานของโปรแกรม

เวลา **t**

Main memory	address	variable
	0	
0000 -1010	1	10 X มีค่า 10
0100 -0001	2	'A' C มีค่า 'A'
0000 0001	3	257 F มีค่า 257
0000-0001	4	

เวลา **t+1 : X := 15**

Main memory	address	variable
	0	
0000 -1111	1	10 X มีค่า 15
0100 -0001	2	'A' C มีค่า 'A'
0000 0001	3	257 F มีค่า 257
0000-0001	4	

ตัวแปร **X** มีขนาด **1** ช่อง อยู่ที่ตำแหน่งของหน่วยความจำช่องที่ **1** จัดเก็บค่า **10**

**C** มีขนาด **1** ช่อง อยู่ที่ตำแหน่งของหน่วยความจำช่องที่ **2** จัดเก็บค่า **10**

**F** มีขนาด **2** ช่อง อยู่ที่ตำแหน่งของหน่วยความจำช่องที่ **3** และ **4** จัดเก็บค่า **257**

**‘:=’** คำสั่ง assignment

# การสร้างตัวแปร (Variable declaration)

- ในภาษา **high lang.** ก่อนการใช้งานตัวแปรต้องเรียกคำสั่ง คือ “กำหนดชนิดตัวแปร (**declare type statement**)” ก่อน
- เพื่อให้ระบบทราบว่าต้องจองหน่วยความจำ (**memory cell**) จำนวนทั้งหมดที่ช่องให้กับตัวแปร

```
int main() {  
    // ตัวแปรชนิด int (จำนวนเต็ม)  
    int age = 25;  
  
    // ตัวแปรชนิด float (จำนวนจริง)  
    float height = 175.5;  
  
    // แสดงผล  
    printf("Age: %d\n", age);  
    printf("Height: %.1f\n", height);  
  
    return 0;  
}
```

C/C++

```
# ตัวแปรชนิด int (จำนวนเต็ม)  
age = 25  
  
# ตัวแปรชนิด float (จำนวนจริง)  
height = 175.5  
  
# แสดงผล  
print("Age:", age)  
print("Height:", height)
```

Python

```
program PersonInfo;  
var  
    age: integer;  
    height: real;  
begin  
    age := 25;           { กำหนดค่าอายุ }  
    height := 175.5;     { กำหนดค่าส่วนสูง }  
  
    writeln('Age = ', age, ' years');  
    writeln('Height = ', height:0:1, ' cm');  
end.
```

pascal

# หน้าที่ของ Data Type บนภาษาโปรแกรม (1/5)

- การระบุจำนวน memory cell ที่ต้องจัดสรรให้ตัวแปร โดยขึ้นอยู่กับ spec ของภาษาโปรแกรมและ compiler

ภาษาโปรแกรม	สถาปัตยกรรม	int	float	หมายเหตุ
C (Turbo C)	16-bit (DOS)	2 byte	4 byte	แบบเก่า, ใช้ int = 16 bit
C (GCC บน 64-bit)	64-bit	4 byte	4 byte	มาตรฐาน modern C
Pascal (Turbo Pascal)	16-bit	2 byte	6 byte (real)	real เก็บทศนิยม 6 byte
Java	32/64-bit	4 byte (int)	4 byte (float)	ขนาดคงที่ ไม่เปลี่ยนตามเครื่อง
Python	64-bit	dynamic	dynamic	ไม่มีขนาดตายตัว, ขึ้นกับค่าจริง
Swift	64-bit	8 byte (Int)	8 byte (Double)	Int = 64-bit platform-dependent
Rust	64-bit	i32 = 4 byte, i64 = 8 byte	f32 = 4 byte, f64 = 8 byte	ต้องระบุชนิดให้ชัด
C#	64-bit	4 byte (int)	4 byte (float)	คงที่

# หน้าที่ของ Data Type บนภาษาโปรแกรม (3/5)

Datatype1.pas

- ระบุ data format ในการ encode ข้อมูลที่จัดเก็บในหน่วยความจำ

	addr	8 bits	value	
	10	0000-0001	1	0x01
	11	0000-0000		
	12	0011-0001	'1'	0x31
	13	0000 0000		
	14	0000-0000	1.0	0x3f800000
	15	1000-0000		
	16	0011-1111		

```
main.pas
1 program Hello;
2 var f:real;
3     i:integer;
4     c:char;
5 begin
6     f := 1.0;
7     i := 1;
8     c := '1';
9     writeln ('f = ', f, ', sizeof(f) = ', sizeof(f) );
10    writeln ('i = ', i, ', sizeof(i) = ', sizeof(i) );
11    writeln ('c = ', c, ', sizeof(c) = ', sizeof(c));
12 end.
```

Character : ASCII , Unicode

DEC	OCT	HEX	BIN	Symbol
49	061	31	00110001	1
50	062	32	00110010	2

Float:

IEEE 754 Floating Point Standard

Integer : Binary value(ฐานสอง)

1 -> 0000-0001

3-> 0000-0011

17-> 0001-0001

# หน้าที่ของ Data Type บนภาษาโปรแกรม (4/5)

Datatype1.pas

- 0x31


addr	8 bits	value
10	0000-0001	1
11	0000-0000	0x01
12	0000-0000	
13	0000-0000	
14	0011-0001	'1' 0x31
15	0000 0000	
16	0000-0000	1.0
17	1000-0000	0x3f800000
18	0011-1111	

```
1 program Hello;
2 var f:real;
3     i:integer;
4     c:char;
5 begin
6   f := 1.0;
7   i := 1;
8   c := '1';
9   writeln('f = ', f);
10  writeln('i = ', i);
11  writeln('c = ', c);
12 end.
```

1 - > integer constant  
'1' - > character constant  
1.0 - > floating point constant

# หน้าที่ของ Data Type บนภาษาโปรแกรม (5/5)

ภาษาโปรแกรมรองรับประเภทข้อมูล(data type) ไม่เหมือนกัน

 Primitive Data Types: C vs Swift		
ประเภทข้อมูล	ภาษา C	ภาษา Swift
Integer	<code>int</code> , <code>short</code> , <code>long</code> , <code>long long</code> (signed/unsigned)	<code>Int</code> , <code>UInt</code> , <code>Int8</code> , <code>UInt8</code> , <code>Int16</code> , <code>Int32</code> , <code>Int64</code>
Floating Point	<code>float</code> , <code>double</code> , <code>long double</code>	<code>Float</code> , <code>Double</code> , <code>Float80</code> (ในบาง platform)
Character	<code>char</code>	<code>Character</code> , <code>UnicodeScalar</code> , <code>String.Element</code>
Boolean	ไม่มี <code>bool</code> จนกระทั่ง C99 ( <code>_Bool</code> หรือ <code>bool</code> ต้อง include <code>&lt;stdbool.h&gt;</code> )	<code>Bool</code>
String	ไม่มี string type แต่ ใช้ <code>char[]</code> หรือ <code>char*</code>	<code>String</code>

# ตำแหน่งของตัวแปรในหน่วยความจำ และ initialization value

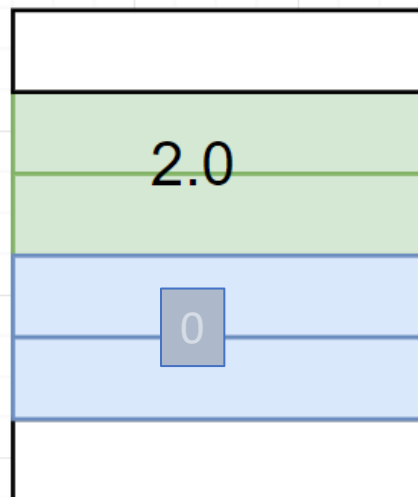
main.pas

```
1 program Hello;
2 var f:real;
3     i:integer; {undefined value }
4 begin
5     f := 2.0; {inital value }
6     writeln ('f = ' , f , ' , addr f = ' , PtrUInt( @f) );
7     writeln ('i = ' , i , ' , addr i = ' , PtrUInt( @i) );
8 end.
```

initvar.pas

Main memory

address



0

1

2

3

4

float f = 2.0 ;

int i ;

Run#1

Run#2

Freepascal

```
Free Pascal Compiler version 3.2.2+dfsg-32 [2024]
Copyright (c) 1993-2021 by Florian Klaempfl and
Target OS: Linux for x86-64
Compiling main.pas
Linking a.out
10 lines compiled, 0.0 sec
f =  2.0000000000000000E+000, addr f = 4384816
i = 0, addr i = 4384832
```



# ตำแหน่งของตัวแปรในหน่วยความจำ และ initialization value

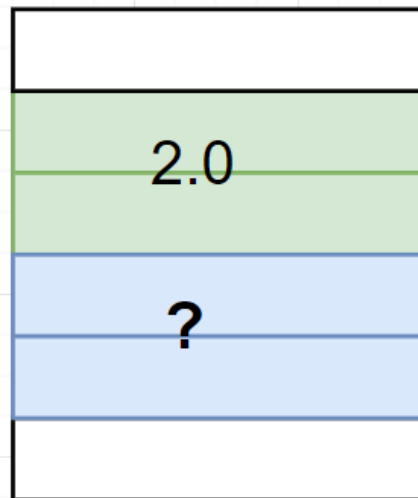
main.cpp

```
1 #include <stdio.h>
2 //initvalue.cpp
3 int main( )
4 {
5     float f= 2.0; //lare f as float
6     int i; //declare i as integer
7
8     printf("\nint i = %d, address = %p , sizeof(i) = %ld", i, (void*)&i,sizeof(i) );
9     printf("\nfloat f = %f, address = %p, sizeof(f)= %ld", f, (void*)&f,sizeof(f) );
10    return 0 ;
11 }
12
```

c/c++

initval.cpp

Main memory



address

0

1

2

3

4

float f = 2.0 ;

int i ;

Run#1

```
int i = 32767, address = 0x7fff681c47e4 , sizeof(i) = 4
float f = 2.000000, address = 0x7fff681c47e0, sizeof(f)= 4

...Program finished with exit code 0
Press ENTER to exit console.[]
```

Run#2

```
int i = 32766, address = 0x7ffeef56b8c4 , sizeof(i) = 4
float f = 2.000000, address = 0x7ffeef56b8c0, sizeof(f)= 4

...Program finished with exit code 0
Press ENTER to exit console.[]
```

# Program Template

```
1
2 program BasicTemplate;      BasTempl.pas
3
4 var
5     { mutable variable }
6     a, b, sum: Integer;
7     name: string;
8
9 begin
10     a := 10;      { initial variable }
11     b := 20;     //initial variable
12
13     sum := a + b;
14     writeln('Sum = ', sum);
15
16
17 {
18     maxInt := 200;
19     maxString := 'new-string'
20 }
21
22 writeln('press any key');
23 readln; { Wait for user before exit }
24 end.
25
```

main.pas

```
1 program <ProgramName>;
2
3 var
4     <identifier_list> : <type>;
5     { more variable declarations if needed }
6
7
8 begin
9     { main program }
10    <statement_list>;
11 end.
12
```

**:= -> Assignment operator**

# Program Template

```
1
2  program BasicTemplate;
3
4  var
5      { mutable variable }
6      a, b, sum: Integer;
7      name: string;
8
9  begin
10     a := 10;      { initial variable }
11     b := 20;  //initial variable
12
13     sum := a + b;
14     writeln('Sum = ', sum);
15
16
17 {
18     maxInt := 200;
19     maxString := 'new-string'
20 }
21
22     writeln('press any key');
23     readln; { Wait for user before exit }
24 end.
25
```

คำศัพท์ของภาษา PASCAL เป็นแบบ insensitive case

sum<-> Sum <-> SUM<-> sUM

writeln <-> WRITELN <-> WriteLN <-> writeLN

program <-> PROGRAM <-> Program

**:= -> Assignment operator**

# Basic data type in (Free)Pascal & Program Template

BasicTypes.pas

```
1 program BasicTypes;
2
3 var
4     age: integer;      { จำนวนเต็ม }
5     height: real;      { จำนวนจริง (ทศนิยม) }
6     grade: char;       { ตัวอักษรหนึ่งตัว }
7     name: string;      { ข้อความหรือสตริง }
8     passed: boolean;   { ค่าตรรกะ true/false }
9
10 begin
11     { กำหนดค่าเริ่มต้นให้ตัวแปรแต่ละตัว }
12     age := 25;
13     height := 175.5;
14     grade := 'A';
15     name := 'Apirak';
16     passed := true;
17
18     { แสดงผลทางหน้าจอ }
19     writeln('Age = ', age);
20     writeln('Height = ', height);
21     writeln('Grade = ', grade);
22     writeln('Name = ', name);
23     writeln('Passed = ', passed);
24 end.
25
```

คำศัพท์ของภาษา PASCAL เป็นแบบ insensitive case

age <-> AGE <-> Age <-> aGE

writeln <-> WRITELN <-> WriteLN <-> writeLN

program <-> PROGRAM <-> Program

**:= -> Assignment operator**

Basic primitive datatype :

Integer -> 12, 102

Real -> 12.0, 102.1

Char -> 'a', 'B'

Boolean -> True, False

String -> 'my name is xxxx'

# Display format

```
1 program FormatDemo;
2 var                                     FormatDemo.pas
3     height: real;
4 begin
5     height := 175.5432;
6     writeln('height: ', height);        { แสดงค่าจริง }
7     writeln('height: ', height:0:2);    { แสดงทศนิยม 2 ตำแหน่ง }
8     writeln('height: ', height:10:2);   { จัดกว้าง 10 ช่อง ทศนิยม 2 ตำแหน่ง }
9 end.
10
```

```
Copyright (c) 1993-2021 by Florian Klaempfl and others
Target OS: Linux for x86-64
Compiling jdoodle.pas
Linking jdoodle
9 lines compiled, 0.1 sec
height:  1.7554320000000000001E+002
height: 175.54
height:    175.54
```

# เทคนิคการค้นหาตัวแปร

myCat.pas

- พิจารณาจาก “ค่านาม”

1. แมวชื่อมีมี เกิดปี 2019 แต่ปัจจุบันปี 2025 อยากรู้ว่าน้องแมวมีอายุเท่าไร?  
myCat bornYear currentYear catAge

myCart มีประเภทข้อมูลคือ String ค่าเริ่มต้น คือ “Memi”

bornYear มีประเภทข้อมูลคือ Int ค่าเริ่มต้น คือ “2019”

CurrentYear มีประเภทข้อมูลคือ Int ค่าเริ่มต้น คือ “2025”

CatAge มีประเภทข้อมูลคือ Int ไม่มีค่าเริ่มต้น ?

main.pas

```
1 program Hello;  
2 var myCat :string;  
3     bornYear:integer;  
4     currentYear:integer;  
5     catAge: integer;  
6 begin  
7     myCat := "Memi";  
8     bornYear := 2019;  
9     currentYear := 2015;  
10 end.  
11  
12
```

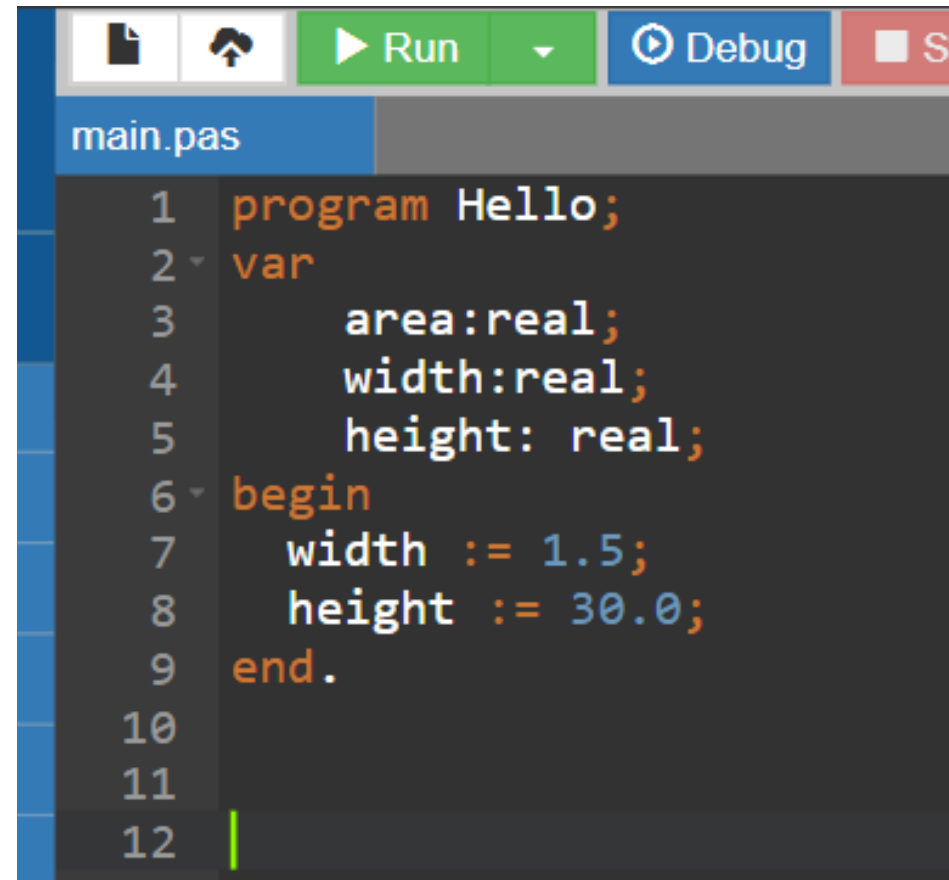
# การสกัดตัวแปร “ค่านาม”

- โปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า จากความกว้างมีค่า **1.5** เมตร และความสูงมีค่า **30** เมตร
- โปรแกรมเก็บชื่อนักเรียน จำนวนหน้า และสถานะว่าอ่านแล้วหรือไม่ (เริ่มต้นอ่านแล้ว = **false**)
- โปรแกรมรับชื่อและอายุของนักเรียน ?
- โปรแกรมคำนวณพื้นที่วงกลม โดยรัศมีมีค่าเป็น 5.0
- โปรแกรมคำนวณค่าเฉลี่ยของนักเรียนจำนวน 3 วิชา โดยแต่ละวิชามีคะแนนดังนี้ 80, 75, 90”
- โปรแกรมเก็บชื่อ อายุ และสถานะผ่านการสอบ
- โปรแกรมคำนวณ **Body Mass Index (BMI)** โดยรับน้ำหนักและส่วนสูง
- โปรแกรมคำนวณความเร็วเฉลี่ย โดยรับระยะทางและเวลา
- โปรแกรมเก็บชื่อผู้เล่น คะแนน และสถานะว่าเป็นผู้ชนะ โดยสถานะมีค่าเป็นจริง
- โปรแกรมรับชื่อและคะแนนสอบวิชาคณิตศาสตร์และวิชาวิทยาศาสตร์ แล้วคำนวณค่าเฉลี่ย
- โปรแกรมเก็บชื่อสินค้า ราคาต่อหน่วย และจำนวนสินค้าในสต็อก (ค่าเริ่มต้นจำนวนสินค้า = 10)

# การสกิดตัวแปร “คำนาม”

- โปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า จากความกว้างมีค่า 1.5 เมตร และความสูงมีค่า 30 เมตร
  - พื้นที่สี่เหลี่ยมผืนผ้า -> area ค่าเริ่มต้นไม่มี
  - ความกว้าง -> width ค่าเริ่มต้น 1.5
  - ความสูง -> height ค่าเริ่มต้น 30

area.pas



```
1 program Hello;
2 var
3     area:real;
4     width:real;
5     height: real;
6 begin
7     width := 1.5;
8     height := 30.0;
9 end.
```



# การสักรัดตัวแปร “ค่านาม”

- โปรแกรมเก็บชื่อนหนังสือ จำนวนหน้า และสถานะว่าอ่านแล้วหรือไม่ (เริ่มต้นอ่านแล้ว = false)
  - ชื่อนหนังสือ : **bookname** ไม่มีค่าเริ่มต้น
  - จำนวนหน้า : **pagenumber** ไม่มีค่าเริ่มต้น
  - สถานะว่าอ่านแล้วหรือไม่ : **readstatus** ค่าเริ่มต้น **false**

bookname.pas

www.onlinegdb.com/online\_pascal\_compiler

main.pas

```
1  program Hello;
2  var
3      bookname:string;
4      pagenumber:integer;
5      readstatus: boolean;
6  begin
7      readstatus := false;
8  end.
9
10
11
```

# การสั้ดตัวแปร “ค่านาม”

- โปรแกรมคำนวณพื้นที่วงกลม โดยรัศมีมีค่าเป็น 5.0
  - พื้นที่วงกลม (area) ไม่มีค่าเริ่มต้น
  - รัศมี (Radiance) ค่าเริ่มต้น **5.0**

radia.pas

main.pas

```
1 program Hello;
2 var
3     area:real;
4     radiance:real;
5 begin
6     radiance := 5.0;
7 end.
8
```

# เทคนิคการค้นหาตัวแปร

- พิจารณาจาก “คำถาม”

1. แมวชื่อมีมี่ เกิดปี **2019** แต่ปัจจุบันปี **2025** อยากรู้ว่ำน้องแมวมียุเท่าไร?
2. น้องมีเงิน **20** บาท ขนมหูละ **5** บาท อยากรู้ว่ำน้องซื้อได้กี่ถุง และเหลือเงินกี่บาท?
3. วิชาคณิตศาสตร์ น้องได้ **60** คะแนน วิชาวิทยาศาสตร์ได้ **75** คะแนน จงหาคะแนนรวม และคะแนนเฉลี่ย?
4. ให้ถามคนข้างๆว่าซื้ออะไร เกิดปีอะไร และปีนี้อายุเท่าไร?
5. รถขับด้วยความเร็ว **60** กิโลเมตรต่อชั่วโมง ถ้าขับเป็นเวลา **1.5** ชั่วโมง จะไปได้กี่กิโลเมตร?

# เทคนิคการค้นหาตัวแปร

- พิจารณาจาก “คำนาม”

- แมวชื่อ มีมี เกิดปี 2019 แต่ปัจจุบันปี 2025 อยากรู้ว่าน้องแมวมีอายุเท่าไร?  
myCat bornYear currentYear catAge
- น้องมีเงิน 20 บาท ขนมถุงละ 5 บาท อยากรู้ว่าน้องซื้อได้กี่ถุง และเหลือเงินกี่บาท?  
myMoney snackPrice totalBags myMoneyLeft
- วิชาคณิตศาสตร์ น้องได้ 60 คะแนน วิชาวิทยาศาสตร์ ได้ 75 คะแนน จงหา คะแนนรวม และ คะแนนเฉลี่ย?  
mathScore scienceScore totalScore averageScore
- ให้ถามคนข้างๆว่าชื่ออะไร เกิดปีอะไร และปีนี้อายุเท่าไร?  
yourName yourBornYear yourAge
- รถขับด้วยความเร็ว 60 กิโลเมตรต่อชั่วโมง ถ้าขับเป็นเวลา 1.5 ชั่วโมง จะไปได้กี่กิโลเมตร?  
carSpeed driveTime driveLength

**END**

# Turbo C

main.c

```
1 #include <stdio.h>
2 //type.c
3 int main() {
4     int i = 1;
5     char c = '1';
6     float f = 1.0;
7     printf("\nint i = %d, hex = 0x%x, address = %p , sizeof(i) = %ld", i, *((unsigned int*)&i), (void*)&i, sizeof(i) );
8     printf("\nchar c = %d, hex = 0x%x, address = %p, sizeof(c) = %ld", c, *((unsigned char*)&c), (void*)&c, sizeof(c) );
9     printf("\nfloat f = %f, hex = 0x%x, address = %p, sizeof(f)= %ld", f, *((unsigned int*)&f), (void*)&f, sizeof(f) );
10    return 0;
11 }
```

Run#1

```
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
main.c:
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

    Available memory 4142480

int i = 1, hex = 0x1, address = FFF4 ; sizeof(i) = 2
char c = 49, hex = 0x31, address = FFF3, sizeof(c) = 1
float f = 1.000000, hex = 0x0, address = FFEE, sizeof(f)= 4
Press any key to continue.
```

&i การอ่าน address ของตัวแปร

sizeof( i ) : คำนวณขนาดตัวแปร

\*((unsigned int\*)&i) : encode in memory

# GCC v 14.x

p/x i  
p/x f  
p/x c

main.c

```
1 #include <stdio.h>
2 //type.c
3 int main() {
4     int i = 1;
5     char c = '1';
6     float f = 1.0;
7     printf("\nint i = %d, hex = 0x%x, address = %p , sizeof(i) = %ld", i, *((unsigned int*)&i), (void*)&i, sizeof(i) );
8     printf("\nchar c = %d, hex = 0x%x, address = %p, sizeof(c) = %ld", c, *((unsigned char*)&c), (void*)&c, sizeof(c) );
9     printf("\nfloat f = %f, hex = 0x%x, address = %p, sizeof(f)= %ld", f, *((unsigned int*)&f), (void*)&f, sizeof(f) );
10    return 0;
11 }
```

Run#1

```
int i = 1, hex = 0x1, address = 0x7ffc60fdb400 , sizeof(i) = 4
char c = 49, hex = 0x31, address = 0x7ffc60fdb3ff, sizeof(c) = 1
float f = 1.000000, hex = 0x3f800000, address = 0x7ffc60fdb404, sizeof(f)= 4
```

i : ค่าของ i

&i : การอ่าน address ของตัวแปร

sizeof( i ) : คำนวณขนาดตัวแปร

\*((unsigned int\*)&i) : encode in memory

Run#

```
int i = 1, hex = 0x1, address = 0x7ffd8974ec30 , sizeof(i) = 4
char c = 49, hex = 0x31, address = 0x7ffd8974ec2f, sizeof(c) = 1
float f = 1.000000, hex = 0x3f800000, address = 0x7ffd8974ec34, sizeof(f)= 4
```