

課程(二) 變數與資料型別

Marius Suzuki

大綱

- 變數 Variables
- 基本資料形別 Basic Data Types
- 常數 Constants
- 格式化輸入輸出 Formatted Input/Output

變數 Variables

什麼是變數？

- 數值可變更的資料
- 類似數學的X跟Y

使用變數的兩大步驟

1. 宣告(Declare)

- a. 定義它的資料類型
- b. 命名
 - i. 不能使用某些關鍵字 (e.g. int, printf)
 - ii. 使用有意義的名字

2. 初始化(Initialize)

- a. 賦值
 - i. 可在「宣告時」或「宣告後」進行

變數 Variables

- 宣告語法:

`<type> name;`

- 範例:

`int a; float i, j; char temp;`

變數 Variables

- 初始化語法:

`<type> name = value;` (宣告時直接賦值)

`name = value;` (宣告後賦值)

- 範例:

`int temp = 10; char c = 'M';`

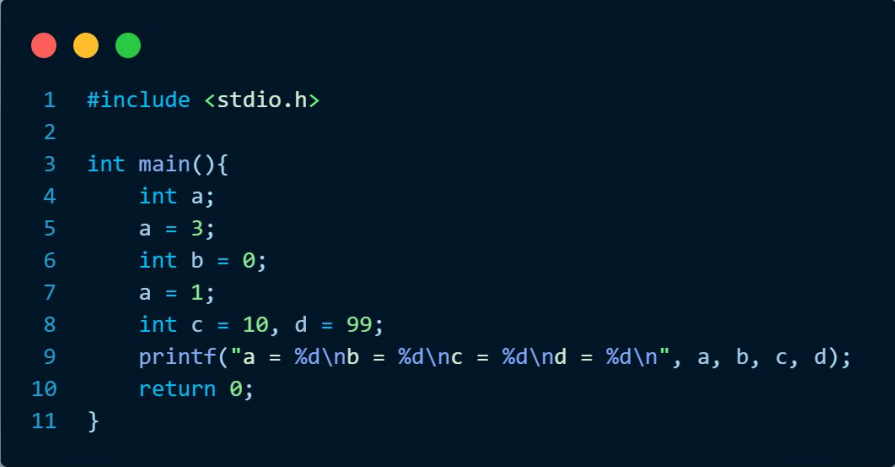
`int i = 0; <some codes...> i = 1;`

基本資料型別表

Name in C	型別	範例/ 解釋	C語言語法	格式指定字
int	整數	7, 0, 3	int i = 3;	%d
float	(浮點數／小數)	1.0, 3.14	float f = 1.0;	%f
long long	長整數	範圍更大的整數	long long ll = 1919810;	%lld
double	精確浮點數	範圍更大、更精準的浮點數	double d = 3.14159265;	%f, %lf
char	字元	一個字。如: 'a', 'A', '?'	char c = 'A';	%c
string	字串	一串字。如: "cat", "Apple"	char name[length] = "Kelly";	%s
bool	布林值	0, 1	bool flag = true;	無

變數 Variables

第9行的輸出結果是什麼？



```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      a = 3;
6      int b = 0;
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

變數 Variables

第9行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4      int a; ←
5      a = 3;
6      int b = 0;
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

a			

變數 Variables

第9行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      a = 3; ←
6      int b = 0;
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

a			
3			

變數 Variables

第9行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      a = 3;
6      int b = 0; ←
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

a	b		
3	0		

變數 Variables

第9行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      a = 3;
6      int b = 0;
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

a	b		
1	0		

變數 Variables

第9行的輸出結果是什麼？

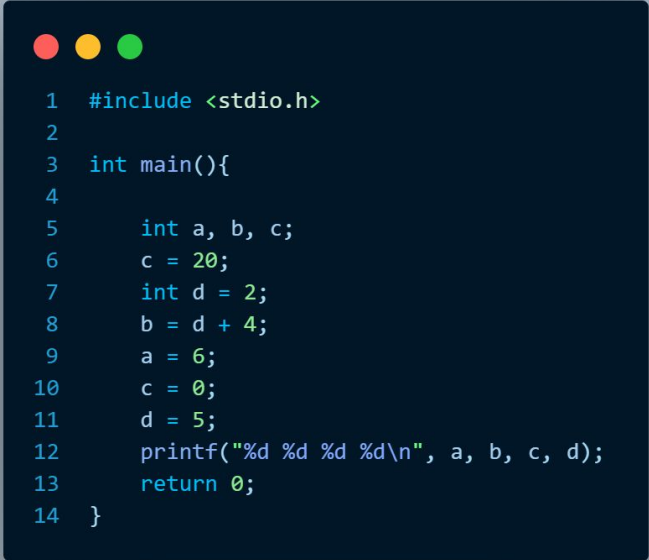
```
1  #include <stdio.h>
2
3  int main(){
4      int a;
5      a = 3;
6      int b = 0;
7      a = 1;
8      int c = 10, d = 99;
9      printf("a = %d\nb = %d\nc = %d\nd = %d\n", a, b, c, d);
10     return 0;
11 }
```

a	b	c	d
1	0	10	99

Any Questions ???

變數

第12行的輸出結果是什麼？



```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```


變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c; ←
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20; ←
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	
		20	

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	d
		20	2

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	d
	6	20	2

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	d
6	6	20	2

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	d
6	6	0	2

變數

第12行的輸出結果是什麼？

```
1  #include <stdio.h>
2
3  int main(){
4
5      int a, b, c;
6      c = 20;
7      int d = 2;
8      b = d + 4;
9      a = 6;
10     c = 0;
11     d = 5;
12     printf("%d %d %d %d\n", a, b, c, d);
13     return 0;
14 }
```

a	b	c	d
6	6	0	5

格式化輸入/輸出

格式化輸出



```
1  #include <stdio.h>
2
3  int main(){
4      printf("Hello nccucs\n");
5      return 0;
6  }
```

格式化輸入/輸出

還記得前面的基本資料型態表嗎

Type	格式指定字 (Format specifier)
int	%d
char	%c
float	%f %n.f
string	%s
long long int	%lld
double	%f, %lf %n.f, %n.lf

格式化輸入/輸出

- 在<stdio.h>裡面的兩個Function
 - `printf()` ;
 - 輸出
 - `scanf()` ;
 - 輸入

格式化輸入/輸出

`printf()`

語法(Syntax):

```
printf("<String...> <format specifier> <String...>", <name...>);
```

範例:

```
printf("My name is %s.\n", my_name);
```

```
printf("I am %s and now I am %d years old.\n", my_name, my_age);
```

格式化輸入/輸出

`scanf()`

語法(Syntax):

```
scanf("<format specifiers...>", &<variables...>);
```

範例:

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

```
scanf("%s %d", country_of_residence, &years);
```

格式化輸入/輸出

`scanf()`

為什麼在變數之前要加一個 “&”？

- 當 & 被加到 Variables 前面的時候表示 (e.g. <name>):
 - 該 Variable 的 Memory Address (記憶體地址)
 - 就像現實的門牌地址一樣
- 需要提供地址，程式才能成功地把使用者輸入資料存入變數

格式化輸入/輸出



```
1  #include <stdio.h>
2
3  int main(){
4
5      char my_name[50]; //string
6      char place_of_residence[50]; //string
7      int years;
8
9      scanf("%s %s %d", my_name, place_of_residence, &years);
10     printf("I am %s from %s. I have been living in Taipei for %d years.\n", my_name, place_of_residence, years);
11
12     return 0;
13 }
```

Any questions ???

格式化輸入/輸出



```
1  #include <stdio.h>
2
3  int main(){
4
5      char my_name[50];
6      int my_age;
7      char college_name[50];
8      double gpa;
9
10     scanf("%s %d %s %lf", _____);
11     printf("I am _____ from the College of _____.\n", _____);
12     printf("I am now _____ and the GPA of my last semester is _____.\n", _____);
13
14     return 0;
15 }
```

格式化輸入/輸出



```
1  #include <stdio.h>
2
3  int main(){
4
5      char my_name[50];
6      int my_age;
7      char college_name[50];
8      double gpa;
9
10     scanf("%s %d %s %lf", my_name, &my_age, college_name, &gpa);
11     printf("I am [ ] from the College of [ ].\n", [ ]);
12     printf("I am now [ ] and the GPA of my last semester is [ ].\n", [ ]);
13
14     return 0;
15 }
```

格式化輸入/輸出



```
1  #include <stdio.h>
2
3  int main(){
4
5      char my_name[50];
6      int my_age;
7      char college_name[50];
8      double gpa;
9
10     scanf("%s %d %s %lf", my_name, &my_age, college_name, &gpa);
11     printf("I am %s from the College of %s.\n", my_name, college_name);
12     printf("I am now [ ] and the GPA of my last semester is [ ].\n", [ ], [ ]);
13
14     return 0;
15 }
```

格式化輸入/輸出



```
1  #include <stdio.h>
2
3  int main(){
4
5      char my_name[50];
6      int my_age;
7      char college_name[50];
8      double gpa;
9
10     scanf("%s %d %s %lf", my_name, &my_age, college_name, &gpa);
11     printf("I am %s from the College of %s.\n", my_name, college_name);
12     printf("I am now %d and the GPA of my last semester is %.2lf.\n", my_age, gpa);
13
14     return 0;
15 }
```

常數及符號常數

What is a constant (常數)?

- 數值不可變更的資料
- 常見例子:
 - π
 - $e \approx 2.718$

常數

- 語法(Syntax):

```
const <type> name = value;
```

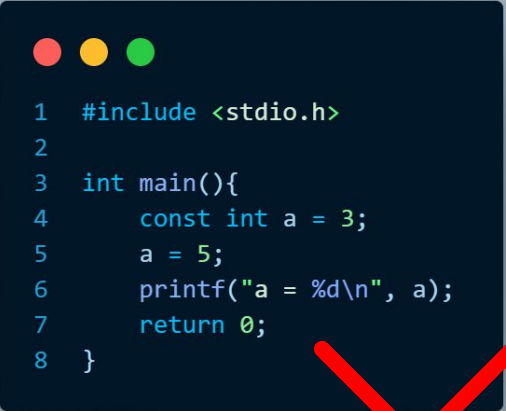
- 範例:

```
const float pi = 3.14;
```

```
const int number_of_students = 40;
```

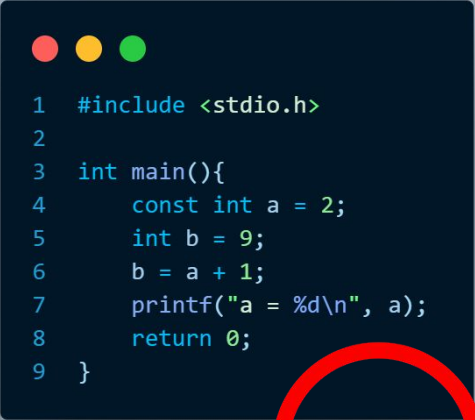
常數

以下三支程式都會編譯成功嗎？



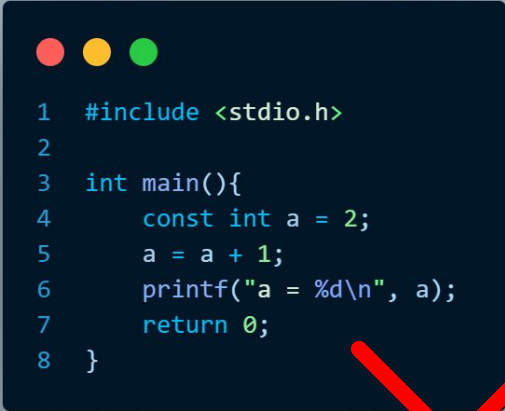
```
1 #include <stdio.h>
2
3 int main(){
4     const int a = 3;
5     a = 5;
6     printf("a = %d\n", a);
7     return 0;
8 }
```

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code defines a constant integer `a` with the value 3 and then attempts to modify its value to 5. A large red 'X' is drawn over the bottom right of the code block.



```
1 #include <stdio.h>
2
3 int main(){
4     const int a = 2;
5     int b = 9;
6     b = a + 1;
7     printf("a = %d\n", a);
8     return 0;
9 }
```

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code defines a constant integer `a` with the value 2 and a regular integer `b` with the value 9. It then increments `b` by 1 and prints the value of `a`. A large red circle is drawn around the bottom right of the code block.



```
1 #include <stdio.h>
2
3 int main(){
4     const int a = 2;
5     a = a + 1;
6     printf("a = %d\n", a);
7     return 0;
8 }
```

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code defines a constant integer `a` with the value 2 and then attempts to modify its value by adding 1. A large red 'X' is drawn over the bottom right of the code block.

另一種實現常數效果的辦法

巨集(Macros)

- 文字替換
- Syntax:

```
#define name value
```

- Example:

```
#define pi 3.14    #define letter 'a'
```

另一種實現常數效果的辦法

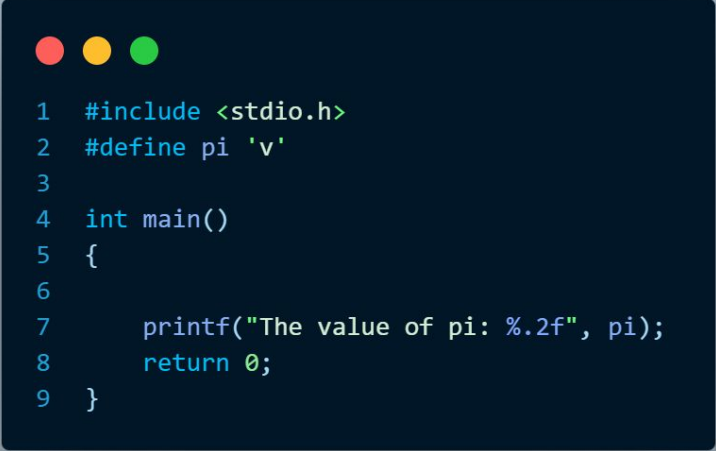
巨集(Macros)

不建議!!!

- 風險
 - 沒有 類別檢查(type checking)
 - 更難除錯(Debug)

另一種實現常數效果的辦法

- 這段程式碼可編譯嗎？為什麼？
- 如果可以的話，你覺得輸出結果會是什麼？



```
1  #include <stdio.h>
2  #define pi 'v'
3
4  int main()
5  {
6
7      printf("The value of pi: %.2f", pi);
8      return 0;
9  }
```

另一種實現常數效果的辦法

- 問題: 類型沒有對上
- 可編譯但Compiler出現警告
- 執行時會出現奇怪的Bugs

```
marius404@Kuri-LAPTOP ~/cs_camp_winter2025/lecture_2 ----- arch x86_64 05:05:16  
> gcc macro.c cpu 0.13 ram 6.76G disk 2% battery 62%  
macro.c: In function 'main':  
macro.c:5:33: warning: format '%f' expects argument of type 'double', but argument 2 has type 'int' [-Wformat=]  
5 |     printf("The value of pi: %.2f", pi);  
  |                               ~~~~^  
  |                               |  
  |                             double  
  |                             %.2d
```

/0.2s

另一種實現常數效果的辦法

Character variables in C

為什麼剛剛編譯器的警告說letter是一個Integer?

除了放字母以外,也可以放數字到Char variables

數字會被當作ASCII Code並翻譯為對應字符

例如: `char start = 65;` → `char start = 'A';`

字元與ASCII CODE

字元與ASCII CODE

Character variables in C

What is ASCII Code?

- 一套以二進制數字表達英文,數字和符號的系統





Why ASCII Code is needed?

- Same as Compiler
- 電腦只會讀0跟1

<https://zh.wikipedia.org/zh-tw/ASCII>

字元與ASCII CODE

marius404@Kuri-LAPTOP ~ ----- arch x86_64 05:00:03

> ascii  cpu 0.06  ram 6.86G  disk 2%  battery 63%

Usage: ascii [-adxohv] [-t] [char-alias...]

-t = one-line output -a = vertical format

-d = Decimal table -o = octal table -x = hex table -b binary table

-h = This help screen -v = version information

Prints all aliases of an ASCII character. Args may be chars, C \-escapes, English names, ^-escapes, ASCII mnemonics, or numerics in decimal/octal/hex.

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex				
0	00	NUL	16	10	DLE	32	20		48	30	0	64	40	@	80	50	P	96	60	`	112	70	p
1	01	SOH	17	11	DC1	33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
2	02	STX	18	12	DC2	34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
3	03	ETX	19	13	DC3	35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
4	04	EOT	20	14	DC4	36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
6	06	ACK	22	16	SYN	38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
7	07	BEL	23	17	ETB	39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
8	08	BS	24	18	CAN	40	28	(56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
9	09	HT	25	19	EM	41	29)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	75	4B	K	91	5B	[107	6B	k	123	7B	{
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	77	4D	M	93	5D]	109	6D	m	125	7D	}
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

/0.0s

字元與ASCII CODE

```
marius404@Kuri-LAPTOP ~ ----- arch x86_64 05:00:06
> ascii -b                               cpu 0.06 ram 6.86G disk 2% battery 63%
0000000 NUL      0010000 DLE      0100000      0110000 0      1000000 @      1010000 P      1100000 `      1110000 p
0000001 SOH      0010001 DC1      0100001 !      0110001 1      1000001 A      1010001 Q      1100001 a      1110001 q
0000010 STX      0010010 DC2      0100010 "      0110010 2      1000010 B      1010010 R      1100010 b      1110010 r
0000011 ETX      0010011 DC3      0100011 #      0110011 3      1000011 C      1010011 S      1100011 c      1110011 s
0000100 EOT      0010100 DC4      0100100 $      0110100 4      1000100 D      1010100 T      1100100 d      1110100 t
0000101 ENQ      0010101 NAK      0100101 %      0110101 5      1000101 E      1010101 U      1100101 e      1110101 u
0000110 ACK      0010110 SYN      0100110 &      0110110 6      1000110 F      1010110 V      1100110 f      1110110 v
0000111 BEL      0010111 ETB      0100111 '      0110111 7      1000111 G      1010111 W      1100111 g      1110111 w
0001000 BS       0011000 CAN      0101000 (      0111000 8      1001000 H      1011000 X      1101000 h      1111000 x
0001001 HT       0011001 EM       0101001 )      0111001 9      1001001 I      1011001 Y      1101001 i      1111001 y
0001010 LF       0011010 SUB      0101010 *      0111010 :      1001010 J      1011010 Z      1101010 j      1111010 z
0001011 VT       0011011 ESC      0101011 +      0111011 ;      1001011 K      1011011 [      1101011 k      1111011 {
0001100 FF       0011100 FS       0101100 ,      0111100 <      1001100 L      1011100 \      1101100 l      1111100 |
0001101 CR       0011101 GS       0101101 -      0111101 =      1001101 M      1011101 ]      1101101 m      1111101 }
0001110 SO       0011110 RS       0101110 .      0111110 >      1001110 N      1011110 ^      1101110 n      1111110 ~
0001111 SI       0011111 US       0101111 /      0111111 ?      1001111 O      1011111 _      1101111 o      1111111 DEL      /0.0s
```

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	~
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[END OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

字元與ASCII CODE

Character variables in C (Extra)

ASCII Code是用於表示英文及數學符號

那其他語言的文字呢？

當然有！！！！

繁體中文(台灣): Big5 (大五碼) 繁體中文(港澳) : Big5 + 香港增補字符集

簡體中文: GB (中華人民共和國國家標準) 日文: JIS (日本產業規格)

涵蓋大部分語言文字: Unicode (UTF-8)

Any questions ???

練習時間

練習

給你一個字串，代表某個城市的名稱；以及一個整數，表示該城市與台北之間的距離(公里)。

請使用上述變數輸出以下內容，最後輸出請空一行。

XXX is YYY Kilometers away from Taipei.

範例：

輸入：Seoul 1483

輸出：Seoul is 1483 Kilometers away from Taipei.

輸入：Oslo 8688

輸出：Oslo is 8688 Kilometers away from Taipei.

END
