Data type

Variable

- A variable has a name or identifier that must begin with an alphabetic letter or the underscore _ character.
  - Example: tong, dem, _abc, i, j, n

- In C variables do not only have names: they also have types
  - Example:
    int i,j;
    char ch;

Where are variables?

```c
#include <stdio.h>
int main()
{
  int tong=0, dem=0, sopt, sosau;
  printf("So phan tu trong day so:");
  scanf("%d", &sopt);

  while (dem < sopt)
  { scanf("%d", &sosau);
    tong += sosau;
    dem++;
  }

  printf("Tong la %d\n",tong);
  return 0;
}
```

Solution

```c
#include <stdio.h>
int main()
{
  int tong=0, dem=0, sopt, sosau;
  printf("So phan tu trong day so:");
  scanf("%d", &sopt);

  while (dem < sopt)
  { scanf("%d", &sosau);
    tong += sosau;
    dem++;
  }

  printf("Tong la %d\n",tong);
  return 0;
}
```

Variable declaration

- Variables in C should be declared at the beginning of a function following the syntax:

  <data type> <variable list>;
  – Example: int i,j; char ch;

              i = j = 0 ;ch = 'A';

- Assignment = puts a specified value into a specified variable

  – Example: int tong=0, dem=0;

Basic data type

- char : a single ASCII character
- short : a short integer (usually 16-bits)
- int : a standard integer (usually 32-bits)
- long : a long integer
- float : a floating point or real number
- double : a long floating point number

Integer numbers

- Integers are stored as binary numbers in memory.
  - Example: The binary number 10100110 corresponds to which number in decimal format?
- There are different integer types in C and they are called char, short, int, long and long long. The difference between these is the size of the integer which either can hold and the amount of storage required for them.
- The unsigned keyword should be used to specify an unsigned integer
  - Example: signed int i; unsigned int u;

Size of Integers

```
Type                Bits        Possible Values
char                  8              -128 to 127
short                16            -32768 to 32767
unsigned short 16                    0 to 65535
int                  32    -2147483648 to 2147483647
long                 32    -2147483648 to 2147483647
unsigned int   32                    0 to 4294967295
long long            64            -9e18 to + 8e18
```

Keep in mind the capacity of each integer type to avoid number overflow

Find value for the calculations

unsigned char a=128;

unsigned char b=128;

unsigned char c = a + b;

- What is the value for c?

Integer constants

- Decimal format:
  - Example: 123456, -123456
- Hexa format (starts with 0x):
  - Example: 0x12AB, 0xFFFF
- Octal format (starts with 0):
  - Example: 0123456
- Note: 123456 and 0123456 are two different numbers

Example

short i=0,j=0;

short a = 0xFFFF

int x,y;

x = y = 123456;

char k = 0xFF;

- What is the value of k? Given that char is a signed integer with the size of 1 byte.

Floating Point Numbers

- There are also long and short floating point numbers in C : float and double

- All the mathematical functions which C can use require double so it is common to use double.

- Only use float if you want to store small floating point numbers.

Size of floating point numbers

```
Type          Bits            Possible Values
float         32         +/- 10E-37  to +/- 10E38
double        64         +/- 10E-307 to +/- 10E308
```

- The nature of real numbers are: they are stored in memory as binary numbers, so it will always have discrete properties that can not be considered as a real number in nature. So when we have a real value in nature as x, the computer will find a real value approximate to x to store it. Therefore, all calculations with real values in the computer are only "approximate". The larger representing size of the real value, the higher accuracy of calculations.

Floating point constants

- Use point:
  - Example: 123.456, 12.456e-2 = 12.456×10^{-2}
- Use scientific syntax:

  Example:
  - double x,y,z; x = 0.1;
  - y = 2.456E5
  - z = 0;

Characters

- A character is stored as a (1 byte) number called character code.
- Example: 'A' = 65, '0' = 48
- English characters are coded using ASCII table
- Use char type to store a character
- The two declarations below are identical:
  - char ch = 'A';
  - char ch = 65; (The ASCII code of the character)

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

16

Control characters

- In the ASCII table, only characters whose code from 32 (space character) are printable characters, all characters from 0 to 31 are used as control characters.
- Example: character code 13 is **new line**; 9 is horizontal tab
- Control characters are put into programs by using a backslash \ and a special character or number. For example:
  - `'\n'` **new line NL (like pressing return)**
  - `'\t'` **horizontal tab HT**
  - `'\0'` **null character**
  - `'\''` **'**
  - `'\\'` ****

Strings

- Groups of char form strings.
- A string must be enclosed by double quote
- Examples:
  - "Hello world!"
  - "Line1\nLine2\nLine3"
  - "I\'m a teacher"

Logic

- In C, every number give also a boolean value, false if the number is equal to zero, true otherwise.
- We normally used
  - 0 for false
  - 1 for true
- Example:

```
int i = 1;
if ( i )
{
    printf("i has true value");
}
```

const declaration

- A variable declaration with the prefix const indicates that the value of that variable never changes in the program.

- Example:
    - const int i = 5;
    - const char c = 'A';
    - const float pi = 3.14;