Biến và biểu thức

Biến - Hằng

- Biến: Là cấu trúc ghi nhớ chứa dữ liệu trong bộ nhớ máy tính
  - Biến ghi nhớ một dữ liệu nào đó gọi là giá trị (value) của biến. Giá trị này có thể thay đổi trong thời gian sử dụng biến.
  - Biến luôn có tên và kiểu dữ liệu của giá trị mà nó ghi nhớ.
  - Biến phải được định nghĩa trước khi sử dụng.
  - Ví dụ: int item;
- Hằng: Là một phần tử của chương trình mà giá trị không bao giờ thay đổi trong suốt quá trình thực hiện chương trình.
  - Ví dụ: const int max = 100;
- Phép gán:
- Sử dụng toán tử =. Một biến đơn nằm bên trái toán tử = và một giá trị nằm bên phải toán tử.
- Nghĩa là chép giá trị bên phải toán tử vào biến.
  - Ví dụ: item = 12;

Cách đặt tên biến

- Danh hiệu: dùng để đặt tên cho các biến, hằng, chương trình con,…
- Không được sử dụng các từ dành riêng của ngôn ngữ C (keywords) để làm danh hiệu như: include, main, using, int, float,…
- Danh hiệu phải bắt đầu bằng một chữ cái hoặc dấu gạch nối _, theo sau bởi các chữ cái, chữ số hoặc dấu gạch nối _.
- Ngôn ngữ C phân biệt chữ hoa chữ thường.
- Ví dụ: sophantu khác Sophantu.

Tầm vực của biến

- Mỗi biến đều có tầm vực của nó.
- Biến phải được khai báo trước khi sử dụng và dựa vào vị trí khai báo của biến ta xác định được tầm vực của biến.
- Có hai loại là biến cục bộ (biến được khai báo trong một hàm nào đó) và biến toàn cục (biến được khai báo không nằm trong bất kỳ hàm nào).
- Trong C, tầm vực của một biến chính là khối lệnh mà nó được khai báo (một khối lệnh là một dãy các khai báo cùng với các câu lệnh được gộp lại trong cặp dấu ngoặc nhọn { }). Nếu nó được khai báo trong một hàm tầm hoạt động sẽ là hàm đó, còn nếu được khai báo trong vòng lặp thì tầm hoạt động sẽ chỉ là vòng lặp đó

Kiểu dữ liệu

- Mô tả loại dữ liệu cho một biến: int x; float y;
- Các loại kiểu dữ liệu cơ bản: int, float, double, char, void
- Mỗi loại có kích thước khác nhau và lưu trữ các giá trị phù hợp. Có thể kiểm tra kích thước của biến bằng lệnh sizeof: sizeof (x)
- Kiểu dữ liệu dẫn xuất (derived data type):
  - Data type modifier + basic data type -> drived data type
  - unsigned int a;
  - short int b;
  - long int c;
  - long double d;

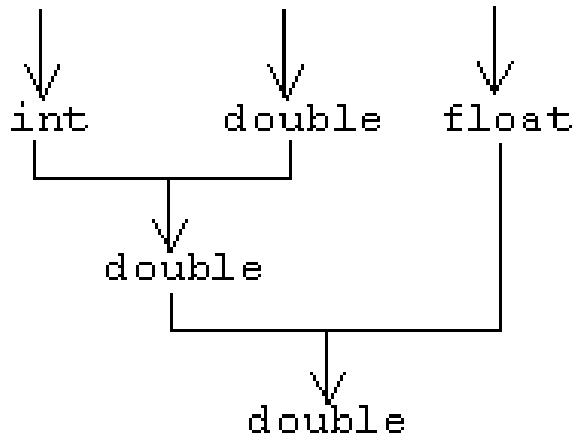
Toán tử và biểu thức

- Toán tử gán: =
- Các phép toán số học: +, -, *, /, %, ++, --, +=, -=, *=, /=, %=
- Các phép toán logical và phép toán quan hệ (relational):
  - Sử dụng so sánh 2 biến hoặc biến và hằng số: >, >=, <, <=, ==, !=
  - Sử dụng kết nối các biểu thức: &&, ||, !
- Các phép toán trên bit (bitwise): Thao tác trên các bit của biến: & (and), | (or), ^ (xor), ~ (not).
  - 10 & 15 -> 1010 & 1111 -> 1010 -> 10
  - 10 | 15 -> 1010 | 1111 -> 1111 -> 15
  - 10 ^ 15 -> 1010 ^ 1111 -> 0101 -> 5
  - ~ 10 -> ~1010 -> 1011 -> -11

Chuyển đổi kiểu dữ liệu

- Chuyển đổi kiểu ngầm định: Khi tính toán 1 biểu thức, ví dụ như các phép toán số học, các hạng tử cần phải có cùng 1 kiểu thì việc tính toán mới diễn ra, nếu các hạng tử không cùng kiểu, sẽ xảy ra chuyển kiểu ngầm định, kiểu nhỏ hơn sẽ chuyển sang kiểu lớn hơn.
- Ví dụ:

```
char ch;
int i;
float f;
double d;
result = (ch/i) + (f*d) - (f+i);
```



Chuyển đổi kiểu dữ liệu

- Chuyển đổi kiểu tường minh: Là việc chuyển 1 dữ liệu sang kiểu dữ liệu khác theo ý muốn.
- Cú pháp:

(<kiểu dữ liệu>)<dữ liệu cần chuyển>

```
int n = 10;
float f1 = (float)n;
float f2 = ((float)10) / 4;
float f3 = (1.0 * 10) / 4;
float f = 10 / 4;
float x=1.23;
int y = (int)x;
int z= x;
```

Lệnh chỉ thị #include, input, output

- Chỉ thị #include: Vị trí: đặt đầu chương trình
- Bộ tiền xử lý (preprocessor) sẽ thực hiện thêm file được chỉ bởi #include vào đầu chương trình
- Cú pháp: #include <header file>

```
#include <stdio.h>
#include <math.h>
```

- Các file header trong C

Header File	Type of Functions
<assert.h>	Diagnostics Functions
<ctype.h>	Character Handling Functions
<locale.h>	Localization Functions
<math.h>	Mathematics Functions
<setjmp.h>	Nonlocal Jump Functions
<signal.h>	Signal Handling Functions
<stdarg.h>	Variable Argument List Functions
<stdio.h>	Input/Output Functions
<stdlib.h>	General Utility Functions
<string.h>	String Functions
<time.h>	Date and Time Functions

Input và output trong C

- Gồm các hàm: printf(), scanf(), putchar(), getchar()
- Nằm trong file header stdio.h
- printf() – sử dụng cho định dạng xuất dữ liệu (formatted output)
- scanf() – for formatted input
- Format: định dạng các giá trị, biến được hiển thị như thế nào.

```
int a, b;
scanf("%d", &a);
scanf("%d", &b);
s = Sum(a, b);
printf("Tong %d + %d=%d", a, b, s);
```

Printf - scanf

- printf và scanf: printf: để xuất dữ liệu ra màn hình và scanf: nhập dữ liệu vào các biến.
- Syntax:
  - printf (“control string”, argument list)
  - scanf (“control string”, argument list)
  - printf(): sử dụng cho tên biến, hằng, biểu thức,…
  - scanf(): Sử dụng cho biến
- control string :
  - 1. Text characters: Những ký tự có thể in
  - 2. Format Commands: % + format code.
  - 3. Ký tự không thể in: (\t) tabs, blanks and (\n) new line

```
#include <stdio.h>
void main()
{
    int n, m;
    printf("Nhap n, m");
    scanf("%d %d", &n, &m);
    printf("[%d]\n",n);
    printf("[%-d]\n",m);
    printf("[%10d]\n",n);
    printf("[%0d]\n",m);
    printf("[%010d]\n",n);
    printf("[%-010d]\n",n);
}
```

Printf-scanf: Control String

Format codes

Format	printf()	scanf()
Single Character	%c	%c
String	%s	%s
Signed decimal integer	%d	%d
Floating point (decimal notation)	%f	%f or %e
Floating point (decimal notation)	%lf	%lf
Floating point (exponential notation)	%e	%f or %e
Floating point (%f or %e , whichever is shorter)	%g	
Unsigned decimal integer	%u	%u
Unsigned hexadecimal integer (uses "ABCDEF")	%x	%x
Unsigned octal integer	%o	%o

Control String Special Characters

\\	to print \ character
\"	to print " character
%%	to print % character

printf- scantf: control string

No	Statements	Control String	What the control string contains	Argument List	Explanation of the argument list	Screen Display
1.	printf("%d",300);	%d	Consists of format command only	300	Constant	300
2.	printf("%d",10+5);	%d	Consists of format command only	10 + 5	Expression	15
3.	printf("Good Morning Mr. Lee.");	Good Morning Mr. Lee.	Consists of text characters only	Nil	Nil	Good Morning Mr. Lee.
4.	int count = 100; printf("%d",count);	%d	Consists of format command only	count	variable	100
5.	printf("\nhello");	\nhello	Consists of nonprinting character & text characters	Nil	Nil	hello on a new line

Prinf - scanf

- Khác biệt format code trong scanf
  - Kiểu dữ liệu cơ bản: Thêm ký hiệu & trước biên.
  - derived data type: Không thêm &
  - Không có %g
  - %f and %e format codes là giống nhau

```
#include <stdio.h>
void main()
{
int a;
float d;
char ch, name[40];
printf("Please enter the data\n");
scanf("%d %f %c %s", &a, &d, &ch,
name);
printf("\n The values accepted are :
%d, %f, %c, %s", a, d, ch, name);
}
```

Getchar() và putchar()

- getchar() – Đọc một ký tự từ keyboard
- putchar() – xuất mọt ký tự lên màn hình

```
#include <stdio.h>
void main()
{
      char letter;
      printf("\nPlease enter any character : ");letter = getchar();
      printf("\nThe character entered by you is %c\n", letter);
      putchar('H'); putchar('\n'); putchar('\t');
      putchar('E'); putchar('\n');putchar('\t'); putchar('\t');
      putchar('L'); putchar('\n');putchar('\t'); putchar('\t'); putchar('\t');
      putchar('L'); putchar('\n');putchar('\t'); putchar('\t'); putchar('\t');
      putchar('\t');putchar('O');
}
```

Bảng mã ASCII (American Standard Code for Information Interchange)