

CẤU TRÚC DỮ LIỆU

Đỗ Thanh Nghị

dtngghi@cit.ctu.edu.vn

NỘI DUNG

- ÔN TẬP NHANH NGÔN NGỮ LẬP TRÌNH C
 - cấu trúc chương trình
 - biến, kiểu, phép toán
 - vào, ra
 - cấu trúc điều khiển
 - kiểu cấu trúc, con trỏ, mảng

CẤU TRÚC CHƯƠNG TRÌNH C

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

```
#include <stdlib.h>
```

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

```
...
```

```
main()
```

```
{
```

```
    /* NNLT C phân biệt ký tự thường hoa */
```

```
    ...;
```

```
    ...;
```

```
}
```

BIẾN, HẰNG, KIỂU, PHÉP TOÁN

- Biến: ký tự bắt đầu phải là alphabet, _

ví dụ:

```
int count;
```

```
float _epsi;
```

- Hằng: 1234, "hello", '1'

ví dụ:

```
#define MY_PI 3.14
```

BIẾN, HẰNG, KIỂU, PHÉP TOÁN

- Kiểu: char, short, int, long, float, double, void, struct, union, mảng, con trỏ, FILE,...

ví dụ:

```
char BETA = '0';
```

```
float avg;
```

```
char filename[50];
```

```
FILE *file;
```

```
char *name = "bob";
```

BIẾN, HẰNG, KIỂU, PHÉP TOÁN

- Phép toán: +, -, *, /, %, ++, --, =, ==, >=, <=, ...

ví dụ:

```
int a = 12, b = 3, c;
```

```
c = a + b;
```

```
c = a % b;
```

```
a++; b--;
```

```
c = a / b;
```

```
if (c == 6) {  
    printf("a ha");  
}
```

VÀO, RA

- Hàm xuất, nhập: printf, scanf, gets, getchar, ...

ví dụ:

```
int a;
```

```
float b;
```

```
char c;
```

```
printf("nhap a:"); scanf("%d", &a);
```

```
printf("nhap b:"); scanf("%f", &b);
```

```
getchar();
```

```
printf("nhap c:"); c = getchar();
```

```
printf("a = %d, b = %f va c = %c \n", a, b, c);
```

CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc điều khiển: if ... else

if (expression)

statement

else if (expression)

statement

else if (expression)

statement

else if ...

else

statement

CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc điều khiển: switch ... case

```
switch (expression) {  
  case const-expr: statements  
  case const-expr: statements  
  ...  
  default: statements  
}
```

CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc điều khiển: for
for (expr1; expr2; expr3)
statement

CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc điều khiển: while, do ... while

`while (expression)`

`statement`

`do`

`statement`

`while (expression);`

KIỂU CẤU TRÚC

- Kiểu cấu trúc: struct

ví dụ:

```
struct point {  
    int x;  
    int y;  
};
```

```
struct point pt;
```

```
printf("%d, %d", pt.x, pt.y);
```

KIỂU CẤU TRÚC

- Mảng, con trỏ cấu trúc

ví dụ:

```
struct point pts[5];
```

```
struct point *p = pts;
```

```
    pts[0].x = 0; pts[0].y = 0;
```

```
    pts[1].x = 1; pts[1].y = 1;
```

```
...
```

```
    printf("pt0(%d, %d)\n", p->x, p->y);
```

```
    printf("pt1(%d, %d)\n", (p+1)->x, (p+1)->y);
```

```
...
```

KIỂU CẤU TRÚC

- Mảng, con trỏ cấu trúc

ví dụ:

```
struct point *p;
```

```
int n = 5;
```

```
int i;
```

```
p = (struct point *)malloc(n*sizeof(struct point));
```

```
for(i=0; i<n; i++) {
```

```
    printf(" nhap toa do diem %d \n", i);
```

```
    printf(" x:"); scanf("%d", &p[i].x);
```

```
    printf(" y:"); scanf("%d", &p[i].y);
```

```
}
```

KIỂU CẤU TRÚC

- Mảng, con trỏ cấu trúc

ví dụ:

```
for(i=0; i<n; i++) {  
    printf("\n Diem %d co toa do", i)  
    printf("  x:%d", p[i].x);  
    printf("  y:%d", p[i].y);  
}
```

KIỂU CẤU TRÚC

- Cấp phát, thu hồi vùng nhớ

ví dụ:

```
struct point *p; int n = 5, new_size;
/* cấp phát với malloc hay calloc */
p = (struct point *)malloc(n*sizeof(struct point));
p = calloc(n, sizeof(struct point));
/* thay đổi kích thước */
p = (struct point *)
    realloc(p, new_size*sizeof(struct point));
/* giải phóng vùng nhớ */
free(p);
```


TÀI LIỆU THAM KHẢO

- Phạm Văn Ất, "*Kỹ Thuật Lập trình C*", NXB Giao thông vận tải, 2006.
- B-W. Kernighan and D-M. Ritchie, "*The C programming Language*", Prentice-Hall, 1988