KIỂU DỮ LIỆU CON TRỎ

❖KHÁI NIỆM

Cho trước kiểu T=<V,O>. Kiểu con trỏ Tp=<Vp,Op> trỏ đến các biến kiểu T. Khi đó, biến kiểu Tp có giá trị là địa chỉ của các biến kiểu T.

- Vp là miền giá trị của kiểu con trỏ Tp gồm giá trị NULL(bằng 0) và các địa chỉ của các biến kiểu T.
- Op là các phép toán trên kiểu con trỏ Tp gồm: tăng địa chỉ (+), giảm địa chỉ (-), phân giải địa chỉ (*), gán giá trị địa chỉ (=).

KIỂU DỮ LIỆU CON TRỎ

❖SỬ DỤNG

- Khai báo kiểu con trỏ:

typedef kiểu_cơ_sở * kiểu_con_trỏ;

- Khai báo biến con trỏ:

kiểu_con_trỏ tên_biến;

hoặc

kiểu_cơ_sở *tên_biến;

Ví dụ:

```
typedef int *IntPtr;
IntPtr a; // tương đương với int *a
```

KIỂU DỮ LIỆU CON TRỎ

❖SỬ DỤNG

- Con trỏ được sử dụng để lưu địa chỉ của biến cấp phát động ⇒ truy xuất biến cấp phát động bằng con trỏ:

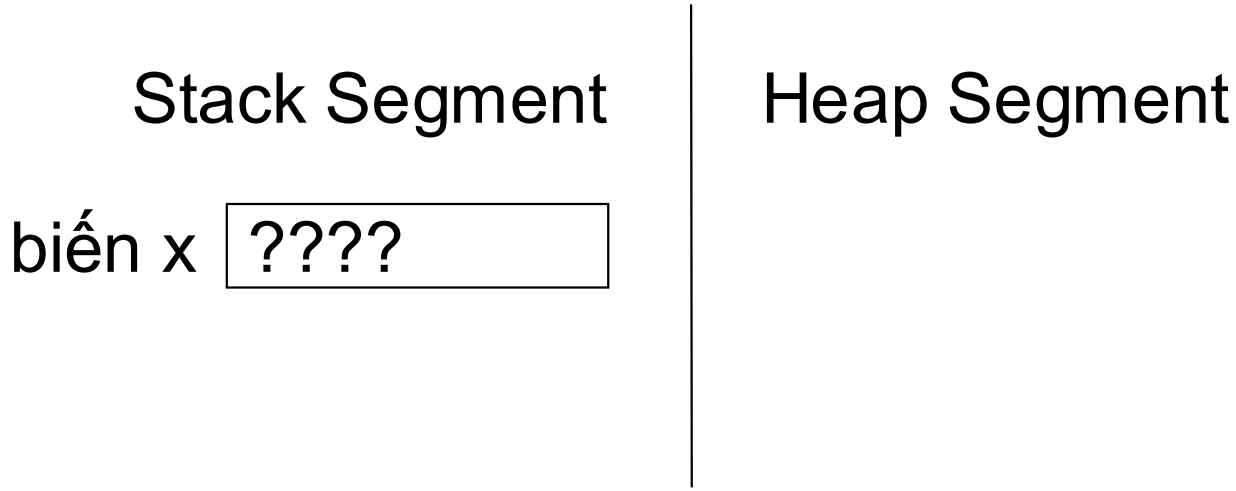
Ví dụ:

```
typedef int *IntPtr;
//....
IntPtr x;
x = new int;
*x = 100;
```

KIỂU DỮ LIỆU CON TRỎ

❖SỬ DỤNG

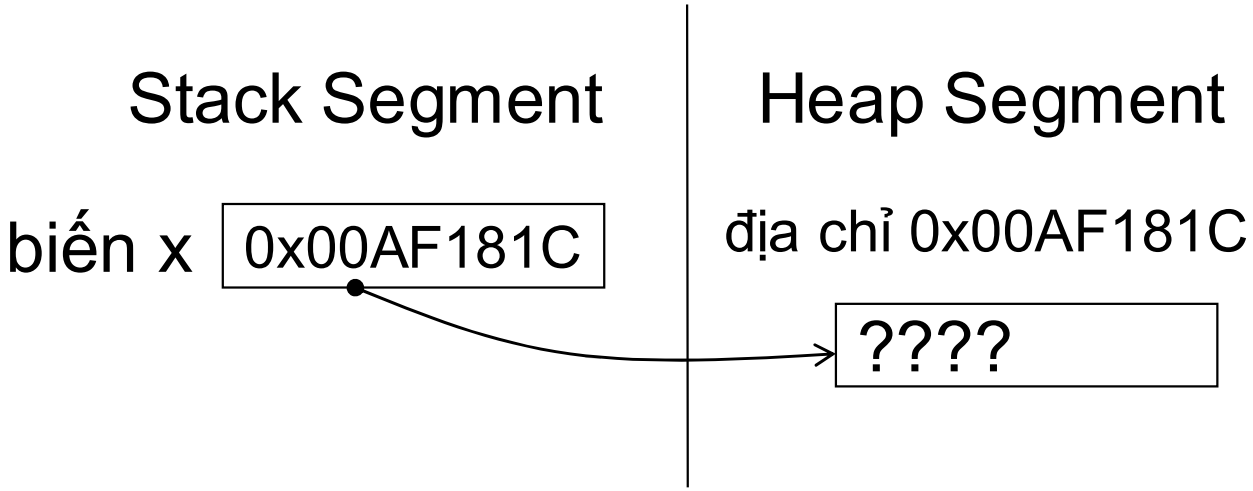
```
typedef int *IntPtr;
//....
IntPtr x;
```



KIỂU DỮ LIỆU CON TRỎ

❖SỬ DỤNG

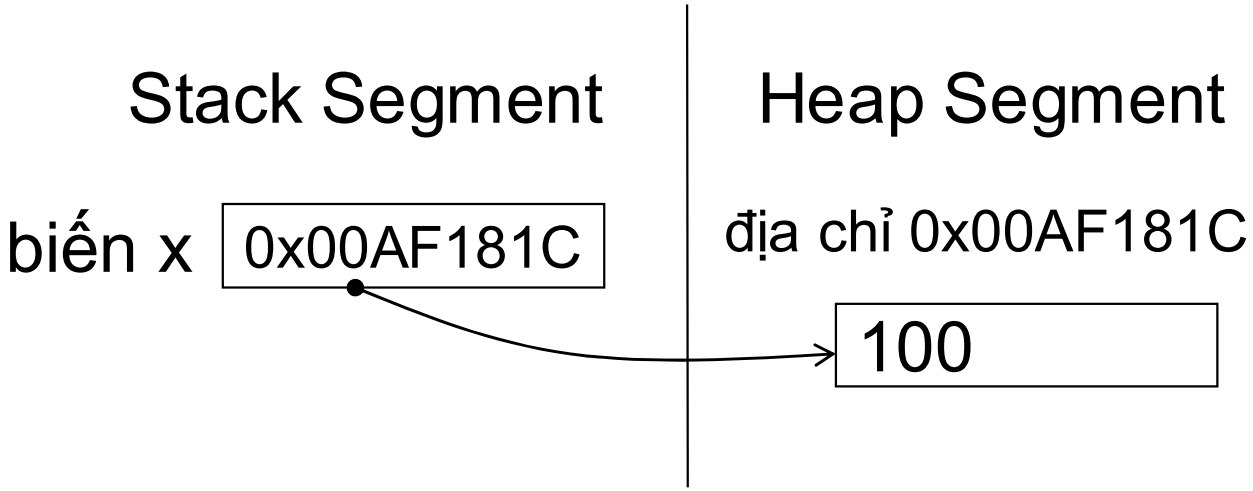
```
typedef int *IntPtr;
//....
IntPtr x;
x = new int;
```



KIỂU DỮ LIỆU CON TRỎ

❖SỬ DỤNG

```
typedef int *IntPtr;
//....
IntPtr x;
x = new int;
*x = 100;
```



Lưu ý: biến con trỏ là một biến cấp phát tĩnh.

KIỂU DỮ LIỆU CON TRỎ

❖CÁC THAO TÁC TRÊN KIỂU CON TRỎ

- Tạo biến cấp phát động để con trỏ quản lý.

+ Trong C, dùng hàm các hàm

void* malloc(size)

void* calloc(n,size)

+ Trong C++, dùng phép toán new

new kiểu // cấp phát vùng nhớ cho 1 biến

new kiểu[n] //cấp phát vùng nhớ cho n biến

Kết quả cấp phát là địa chỉ ô nhớ đầu tiên của vùng nhớ được cấp phát hoặc giá trị NULL nếu không cấp phát được

KIỂU DỮ LIỆU CON TRỎ

❖CÁC THAO TÁC TRÊN KIỂU CON TRỎ

- Giải phóng biến cấp phát động do con trỏ quản lý.

+ Trong C, dùng hàm

```
void free(p)
```

+ Trong C++, dùng phép toán delete

```
delete p          // nếu p = new kiểu
delete [] p       // nếu p = new kiểu[n]
```

KIỂU DỮ LIỆU CON TRỎ

❖CÁC THAO TÁC TRÊN KIỂU CON TRỎ

- Tăng hoặc giảm địa chỉ vùng nhớ do con trỏ quản lý n lần kích thước kiểu: dùng phép toán tương ứng là + hoặc -.

Ví dụ:

```
int *p;
p = (int *)1;
p = p + 4;   // giá trị của p là 1+4*4=17. 1 int = 4 bytes
p--;         // giá trị của p là 17-1*4=13.
```

Lưu ý: *(p + i) tương đương với p[i]

KIỂU DỮ LIỆU CON TRỎ

❖CÁC THAO TÁC TRÊN KIỂU CON TRỎ

- Phân giải địa chỉ con trỏ (dereference) bằng phép toán *. Phân giải địa chỉ con trỏ là truy cập đến biến cấp phát động.

Ví dụ:

```
int *p;
p = new int;      // tạo một biến cấp phát động cho p
*p = 100;         // gán 100 cho biến cấp phát động
*p *= 2;          // nhân 2 với giá trị biến cấp phát động
cout << *p;       // in ra giá trị biến cấp phát động
```

KIỂU DỮ LIỆU CON TRỎ

❖CÁC THAO TÁC TRÊN KIỂU CON TRỎ

Ví dụ: Viết chương trình nhập vào dãy số gồm n giá trị nguyên (n < 1000) và in ra màn hình các giá trị đó theo thứ tự tăng dần.

KIỂU DỮ LIỆU CON TRỎ

```
#include <iostream>
#include <cstdlib>
using namespace std;
typedef int *DaySo;
void BubbleSort(DaySo A, int n) {
    int i, j, x;
    for (i = 0; i < n - 1; i++)
        for (j = n - 1; j > i; j--)
            if (A[j] < A[j-1])
                {x = A[j]; A[j] = A[j - 1]; A[j-1] = x;}
}
```

KIỂU DỮ LIỆU CON TRỎ

```
int main(char **arg, int c) {
    int n, i;
    DaySo A;
    cin >> n;
    A = new int[n];
    if (A == NULL) return EXIT_FAILURE;
    for (i = 0; i < n; i++)
        cin >> A[i];
    cout << endl;
    BubbleSort(A, n);
```

KIỂU DỮ LIỆU CON TRỎ

```
    for (i = 0; i < n; i++)
        cout << A[i] << ' ';
    cout << endl;
    delete [] A;
    return EXIT_SUCCESS;
}
```