

Address

เขียนการชี้ไปที่หน่วยความจำในโปรแกรม

หน่วยความจำ: หน่วยของ บิต เป็น int \rightarrow 4 byte
char 1 ตัว \rightarrow 1 byte

การชี้หน่วยความจำคือ การชี้หน่วย memory address ไปที่หน่วย

Address ของ array เป็น หน่วย ที่อยู่ของ

pointer

เป็น int \rightarrow 4

char 1 ตัว \rightarrow 1

การประกาศหน่วยความจำ

```
int x = 10;
```

```
int *ptr; // pointer
```

ถ้าหน่วยความจำ ptr เก็บค่า x

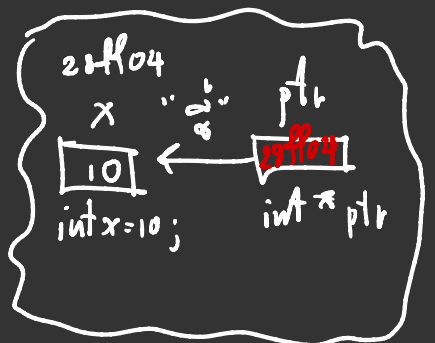
```
ptr = &x
```

```
ptr = 288804 // ค่าหน่วยความจำ
```

กำหนดค่าหน่วยความจำ

```
x = 25;
```

```
ptr = address
```



note: unit address $\%x$, $\&x$

print: `ptr` \rightarrow address of `x`

`*ptr` \rightarrow " " `ptr` // pointer เก็บที่หน่วยความจำ

feel like Hacker, change value

- ถ้าหน่วยความจำ เก็บหน่วยความจำอื่นที่ ไม่ `ptr`.

ptr with array

example program

ptr with array

a+b by pointer

project ptr with pointer

note: print (*ptr) =, in memory address ptr =

if pointer address print

note: */ comment

ptr var 1

var 1



As



ptr1 = (int*) malloc (sizeof(int))

// ptr1 = address

ptr1 = address

ptr1 = address

note: #include <stdlib.h>, c lang

free (ptr1); // delete in c++

malloc in c++

ptr1 = new int; , free is delete in c++

Struct : structure

struct product {
int weight;
float price;
};

product



ကလေးတစ်ခု

```

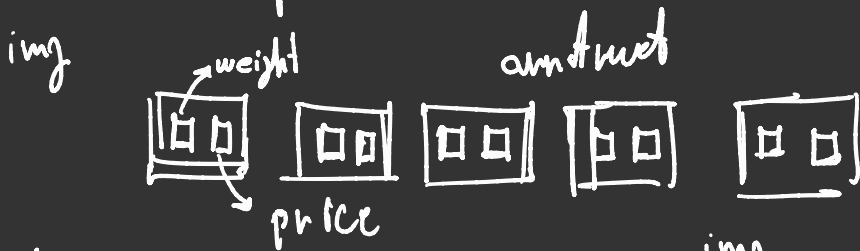
struct product {
    name; // int x; // in
    apple. weight = 10; // x = 10;
    apple. price = 3.14;
}
    
```

အကယ်၍ နံပါတ်များကို ထည့်သွင်းရန်အတွက် array, pointer နှင့် normal နှစ်ခုပါ

C++ note: product name ၏အတွက်
 struct with array

```

struct product arr[5];
    
```



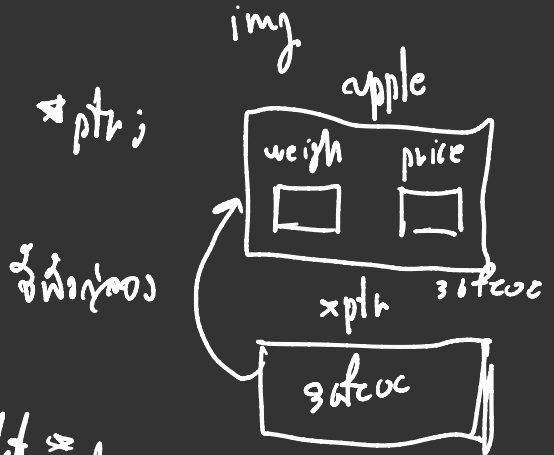
struct with pointer

```

struct product
    
```

```

ptr = &apple;
    
```



အကယ်၍ print

```

printf("%d, %f", *ptr.weight, *ptr.price); // error
    
```

(*ptr).weight, ptr -> weight

Because ptr -> weight အပေါ်မှာ *ptr နှင့် ptr -> weight နှစ်ခုလုံးပါ

struct with pointer (malloc)

ptr = (struct product*, malloc (sizeof (struct product)))

// ptr = new product // C++

change var

ptr -> weight = 50; // (*ptr).weight = 50;

ptr -> price = 0.18;

link list



ကလေးကလေး အားပေးပါ။ လုပ်ပေးပါ။ loop ကို အားပေးပါ။ 1000 ကို အားပေးပါ။ ?

note: queue : ကလေးကလေး ခွေ ~~ခွေ~~ ခွေ ခွေ

stack : ခွေ ခွေ ခွေ ခွေ

ကလေးကလေးကလေး queue ကလေးကလေး

link list : pointer + struct, ခွေ ကလေး 0 ကလေး

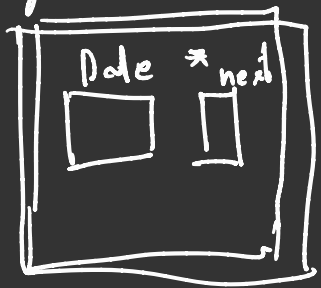
struct node {

int Data;

struct node * next;

};

img



or

```

ex:
#include _____
struct node {
  int data
  struct node *next;
};

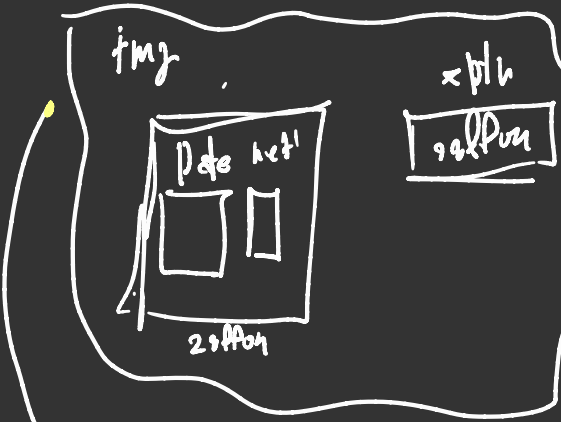
```



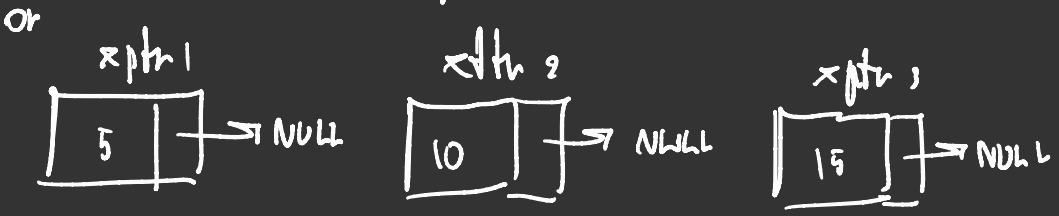
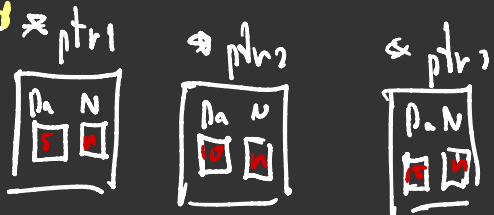
```

int main() {
  struct node *ptr1, *ptr2, *ptr3;
  ptr1 = (struct node *) malloc (sizeof (struct node));
}

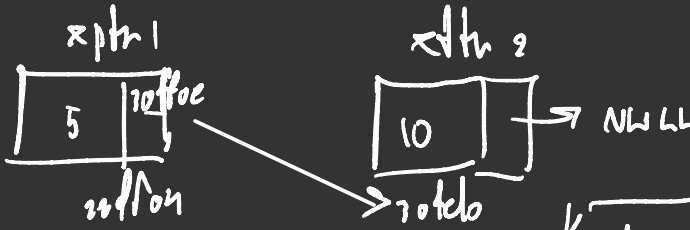
```



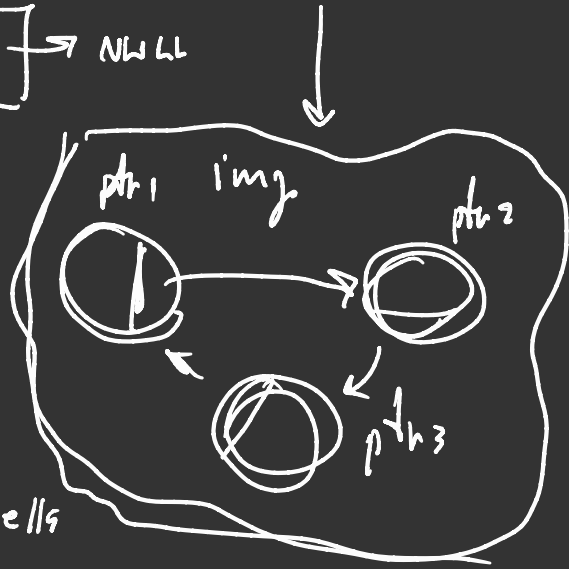
- ptr1 -> data = 5; (ptr1->data)
- ptr1 -> next = NULL;
- ptr2 = (struct node *) malloc (sizeof (struct node));
- ptr2 -> data = 10;
- ptr2 -> next = NULL;
- ptr3 = (struct node *) malloc (sizeof (struct node));
- ptr3 -> data = 15;
- ptr3 -> next = NULL;



ptr1 -> next = ptr2 ; new line ptr2 -> next = ptr3 ;
 ptr3 -> next = ptr1 ;



print, ptr1 -> data // 5
 ptr1 -> next -> data // 10
 ptr1 -> next -> next -> data // 15
 ptr1 -> next -> next -> next -> data // 15

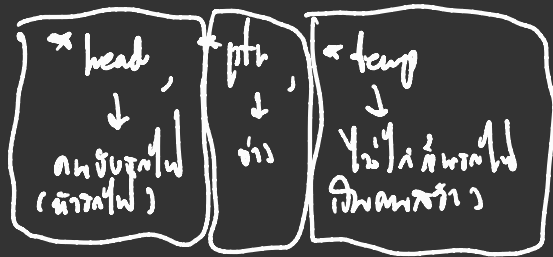


note: ...

minimizing link list in 100 us 1 sum: ...

3 pointer 3 ...
 struct node

step 1



head = struct node*, malloc (- -)

head -> next = NULL

ptr = head // ...

// step 2

tmp = (struct node*) malloc (sizeof struct node);

tmp->data = 5; // insert 5 at the end of the list

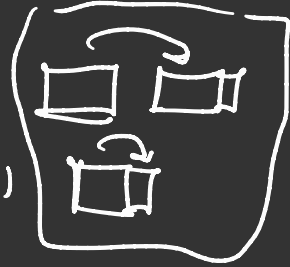
tmp->next = NULL; // end of the list

// step 3

ptr->next->tmp // insert at the end

ptr = ptr->next; // move ptr to tmp (the)

movement



1st loop

// step 2

tmp = (struct node*) malloc (sizeof struct node);

tmp->data = 6; // insert 6 at the end of the list

tmp->next = NULL; // end of the list

// step 3

ptr->next->tmp // insert at the end

ptr = ptr->next; // move ptr to tmp (the)

note: if you want to print next...

ptr = head->next;

while (ptr != NULL) { // print the linked list

```
printf("%d", ptr->data);
```

```
}
```

```
// input n is
```

```
int n;
```

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

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

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

```
    // step 2  
    // step 3
```

```
}
```

note link list min

source code after class

By Neighborhood Cat
in Discord

Note By: Minari Ikkyasan