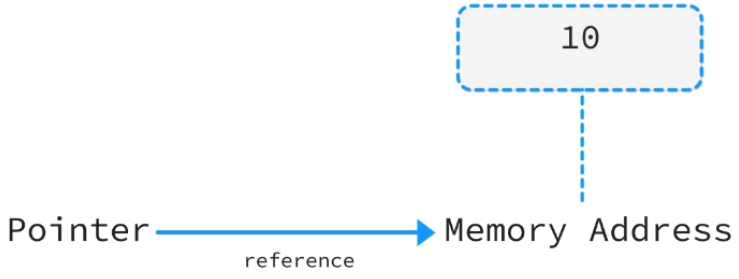
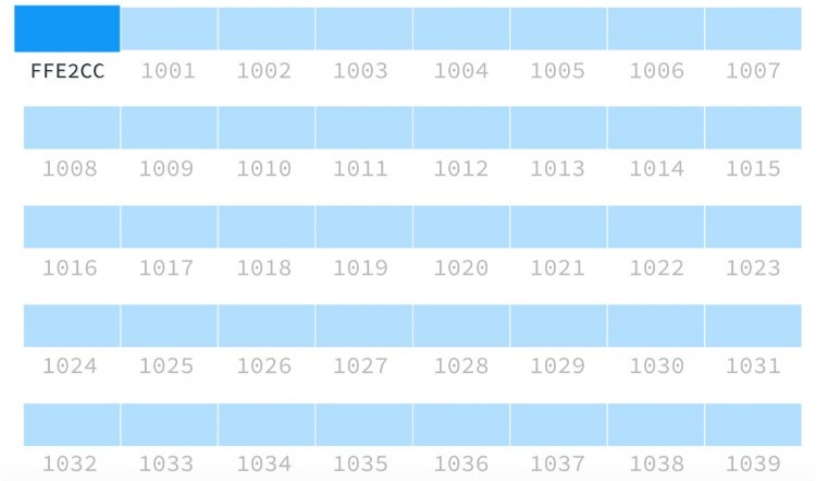
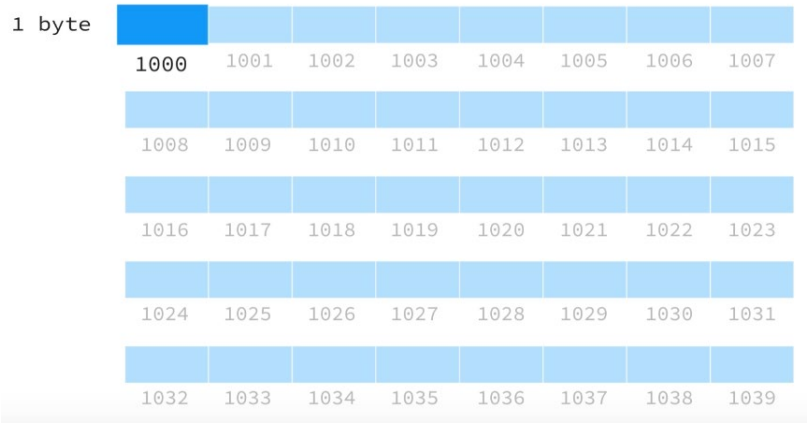
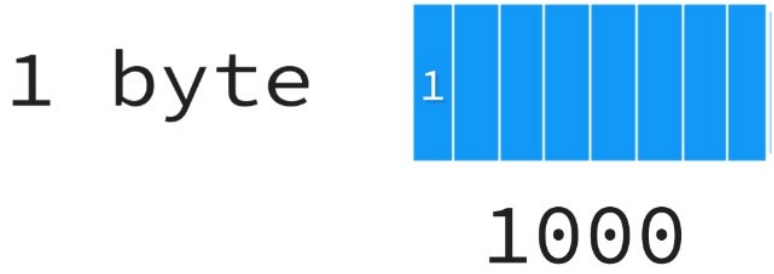
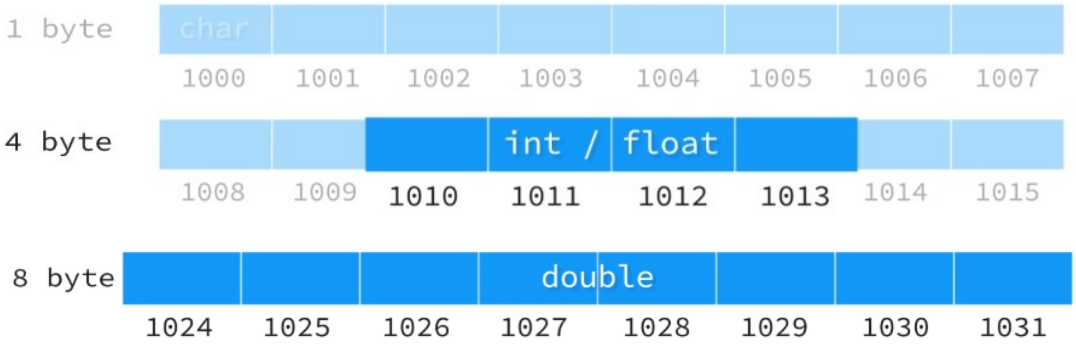
3 记住一句话，地址就是指针，指针就是地址。





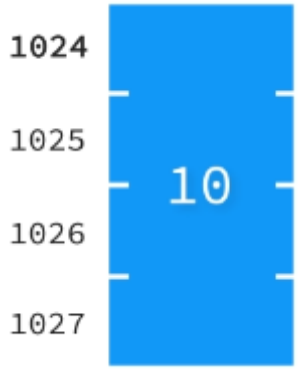




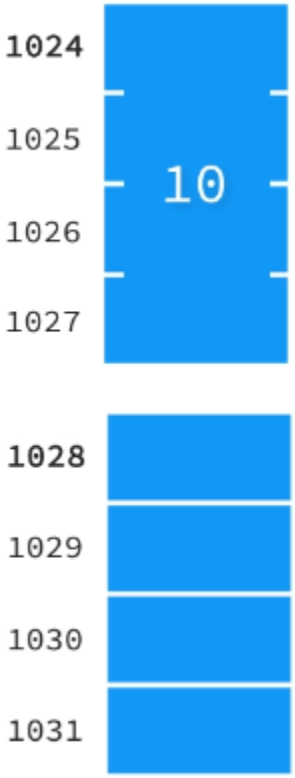



```
int var = 10;
printf("%d",&var);
printf(”%d”,&var+1);
```

所以第一个是地址 1024。



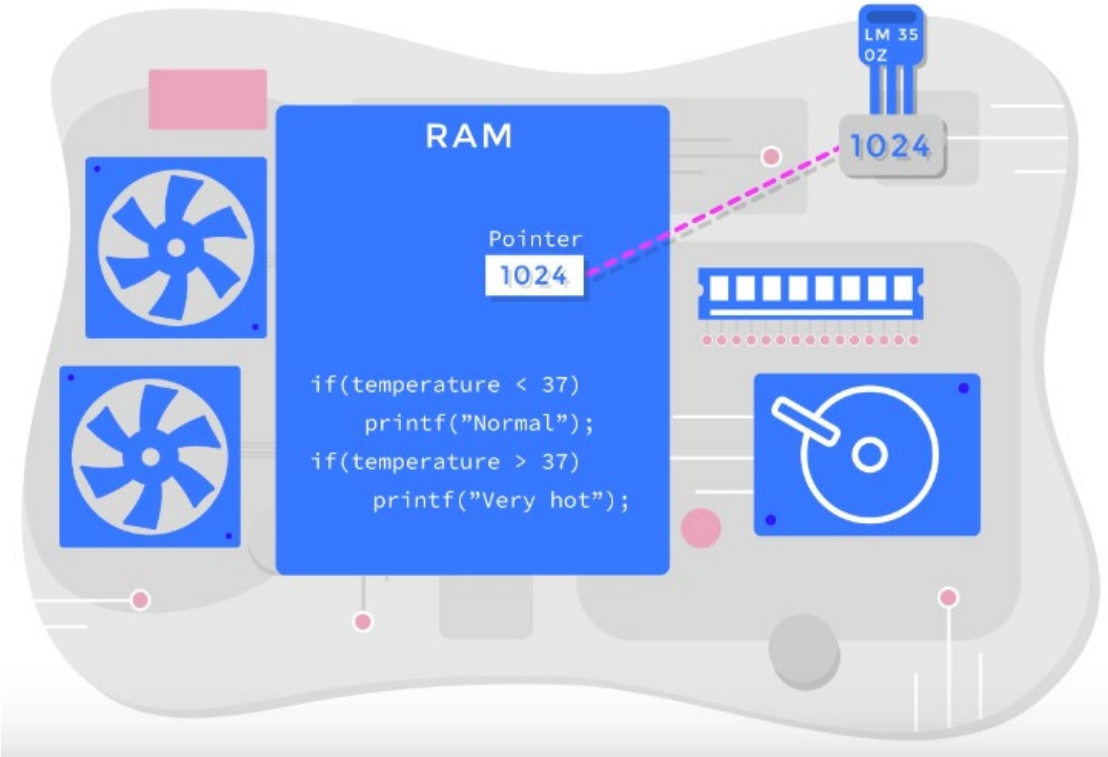
但是打印的第二个值为 1028。

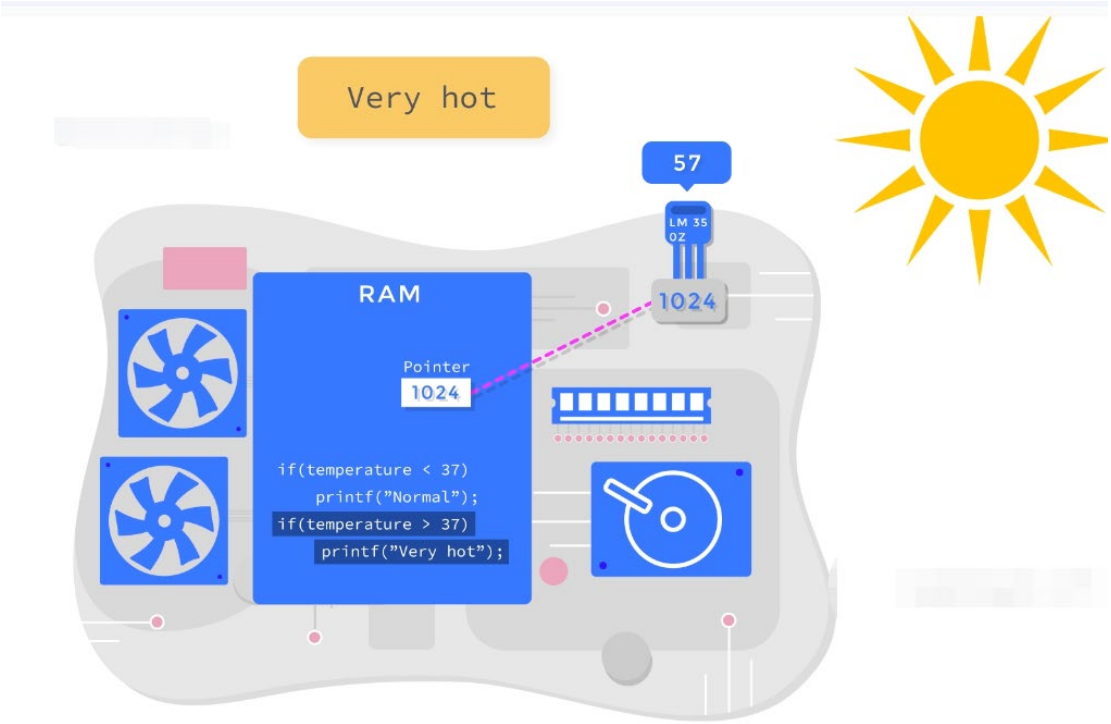


&var+5 表示从当前位置开始的第 5 个字节。

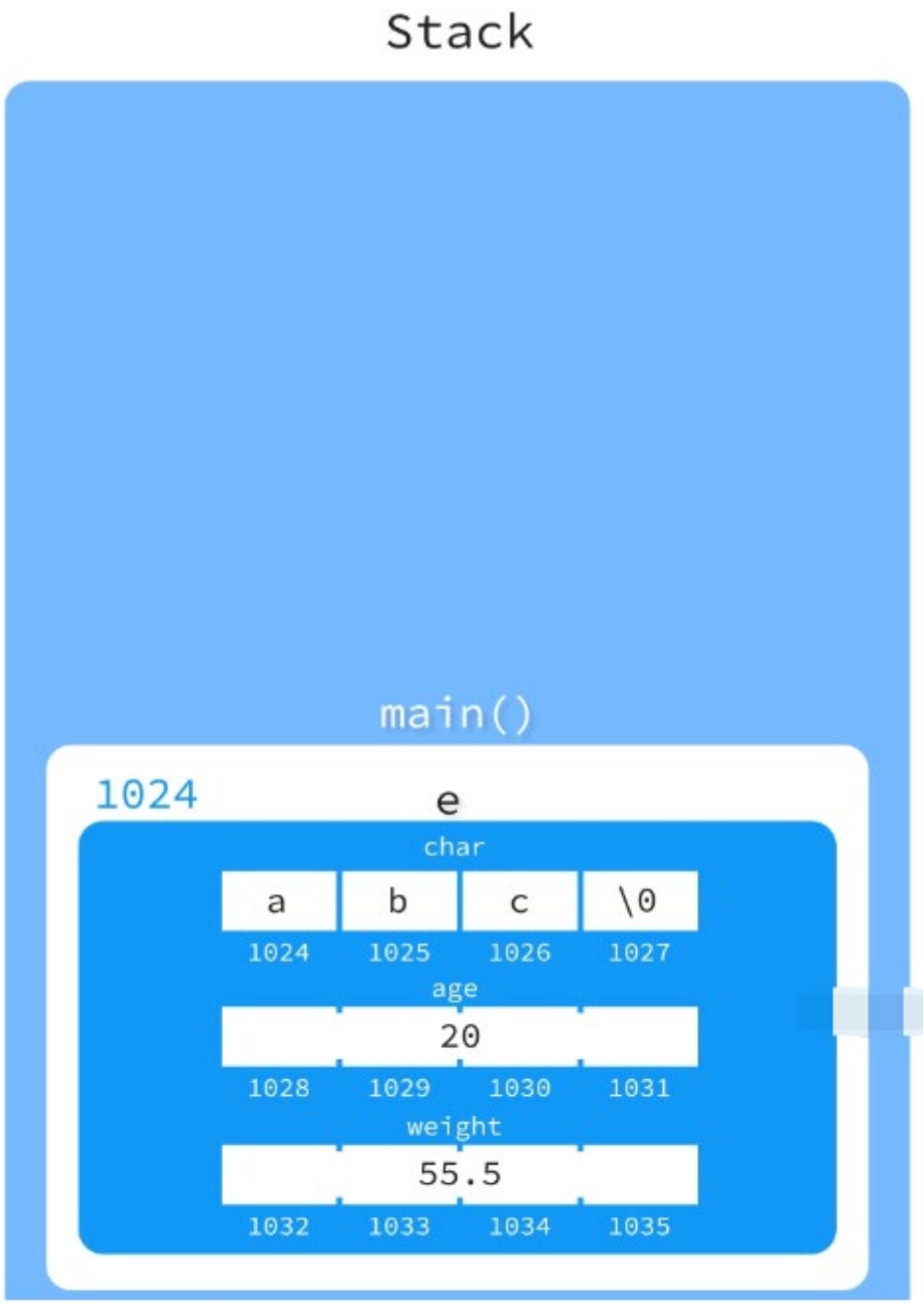
```
&var+5

1024 + 5 * sizeof(int)
```

为此最终的输出是 1024,1028。



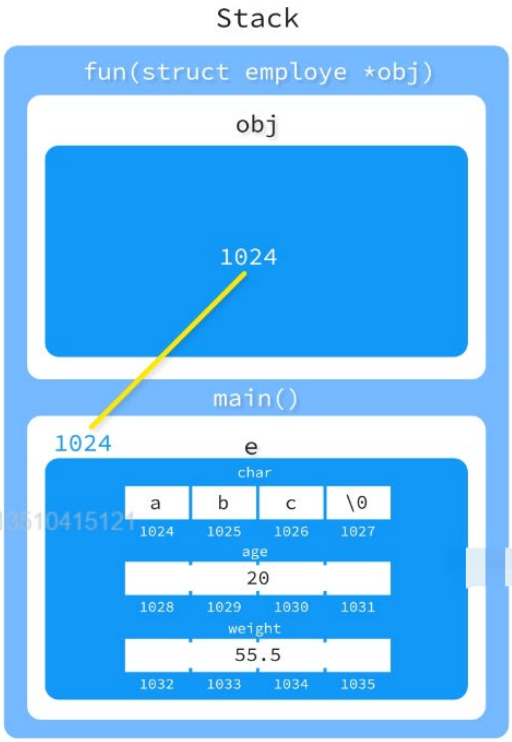

Very hot
57
LM 35
DZ
1024
RAM
Pointer
1024
if(temperature < 37)
printf("Normal");
if(temperature > 37)
printf("Very hot");

```
struct employee
{
    char name[4];
    int age;
    float weight;
};

void fun(struct employee);

int main()
{
    struct employee e = {"abc", 20, 55.5};
    fun(e);

    return 0;
}

void fun(struct employee obj)
{

}
```


Stack
main()
1024
e
char
a
b
c
\0
1024
1025
1026
1027
age
20
1028
1029
1030
1031
weight
55.5
1032
1033
1034
1035

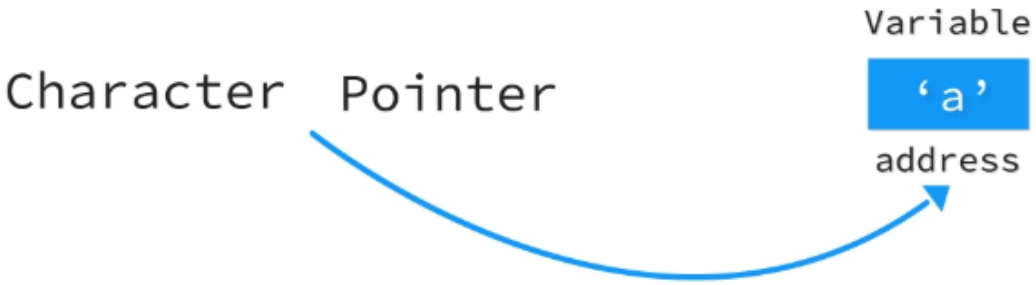

Stack
fun(struct employe obj)
2024
obj
char
a
b
c
\0
2024
2025
2026
2027
age
20
2028
2029
2030
2031
weight
55.5
2032
2033
2034
2035
main()
1024
e
char
a
b
c
\0
1024
1025
1026
1027
age
20
1028
1029
1030
1031
weight
55.5
1032
1033
1034
1035

```
struct employee
{
    char name[4];
    int age;
    float weight;
};

void fun(struct employee*);

int main()
{
    struct employee e = {"abc", 20, 55.5};
    fun(&e);

    return 0;
}

void fun(struct employee *obj)
{

}
```

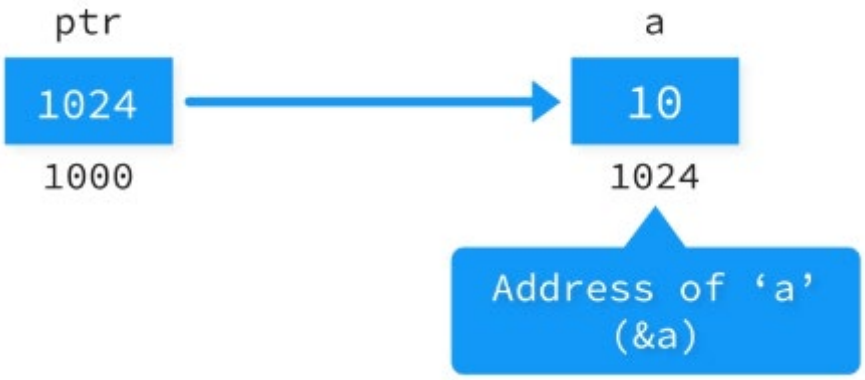


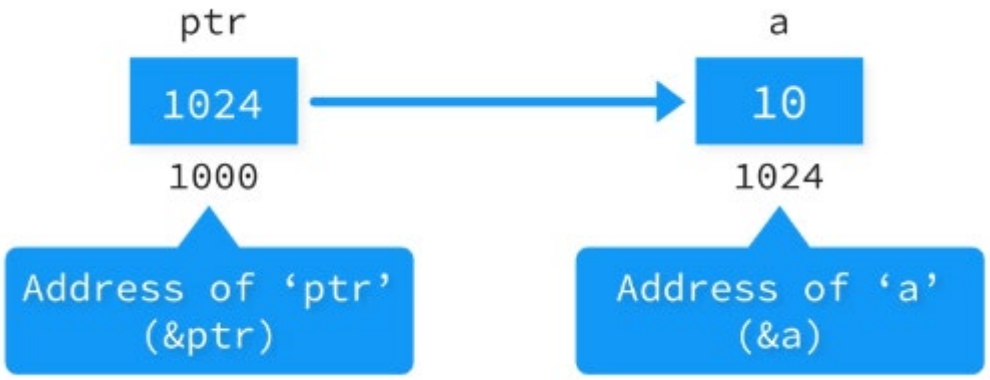
Syntax

```
data_type *pointer_name;
```

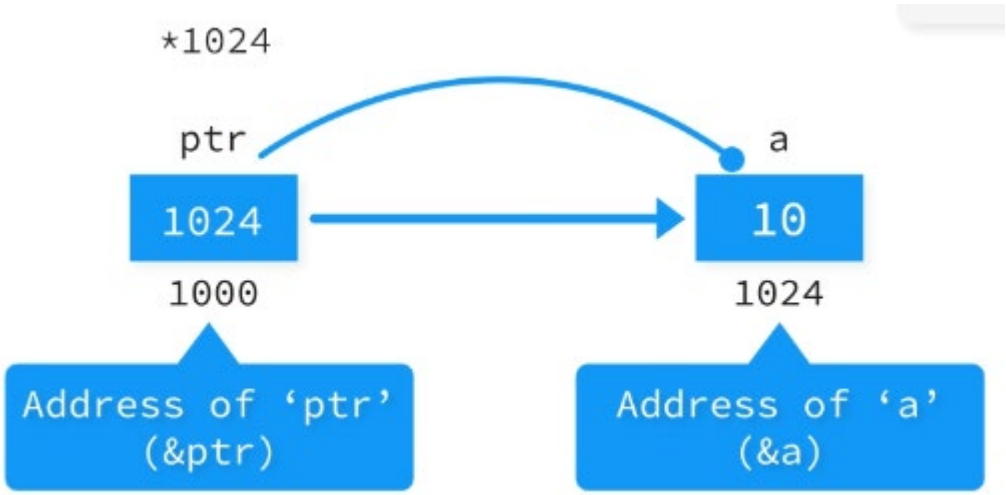



```
int a = 10;
int *ptr = &a;
```





```
int a = 10;
int *ptr = &a;
printf("value of a = %d\n",a);
printf("value stored at ptr = %d\n",*ptr);
```



Output

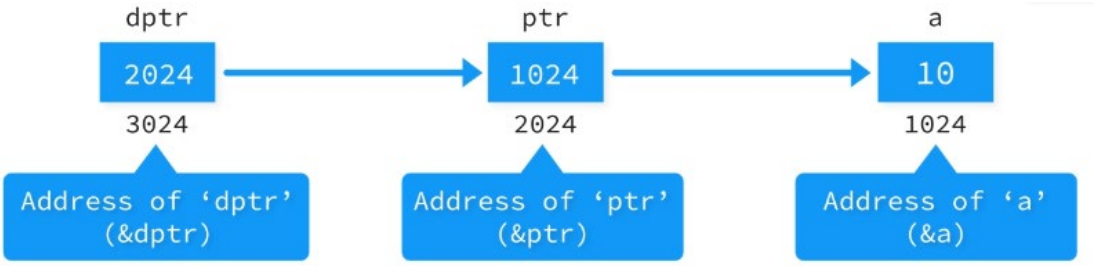
```
value of a = 10
value stored at ptr = 10
```

```
int a = 10;
int *ptr = &a;
printf("value of a = %d\n",a);
printf("value stored at ptr = %d\n",*ptr);
printf("Address of a = %d\n",&a);
printf("ptr points to the address = %d\n",ptr);
printf("Address of ptr = %d\n",&ptr);
```

Output

```
value of a = 10
value stored at ptr = 10
Address of a = 1024
ptr points to the address = 1024
Address of ptr = 1000
```

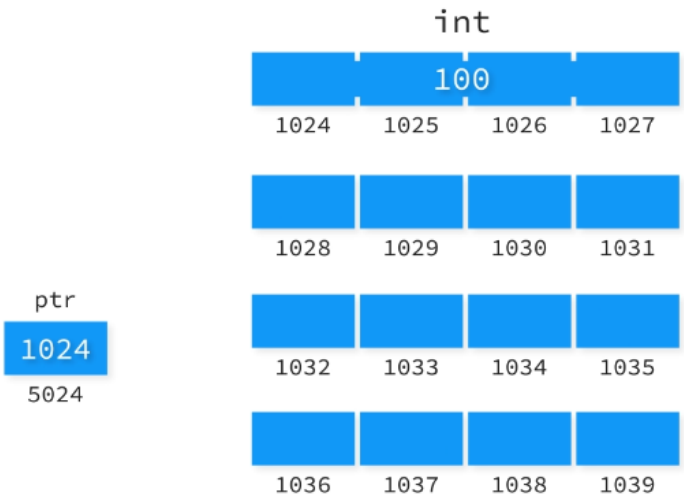
```
int a = 10;
int *ptr = &a;
int **dptr = &ptr;
printf("Address of a = %p\n",&a);
printf("ptr is pointing to the address = %p\n",ptr);
printf("dptr is pointing to the address = %p\n",dptr);
printf("Value of a = %d\n",a);
printf("*ptr = %d\n",*ptr);
printf("**dptr = %d\n",**dptr);
```



Output

```
Address of a = 1024

ptr is pointing to the
address = 1024

dptr is pointing to the
address = 2024

Value of a = 10

*ptr = 10

**dptr = 10
```

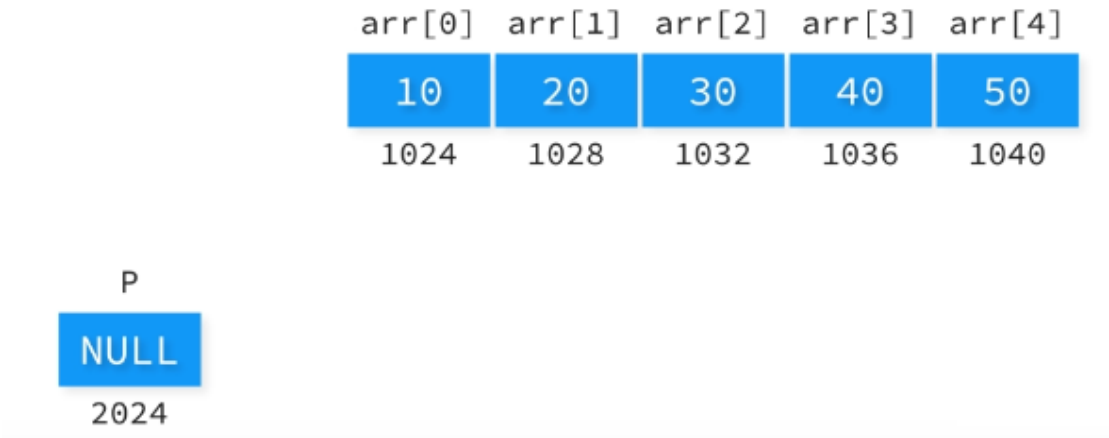
```c
int main()
{
    int i = 100;
    int *ptr = &i;

    printf("ptr   = %p\n",ptr);
    printf("ptr+1 = %p\n",ptr+1);
    printf("ptr+3 = %p\n",ptr+3);

    return 0;
}
```



Output

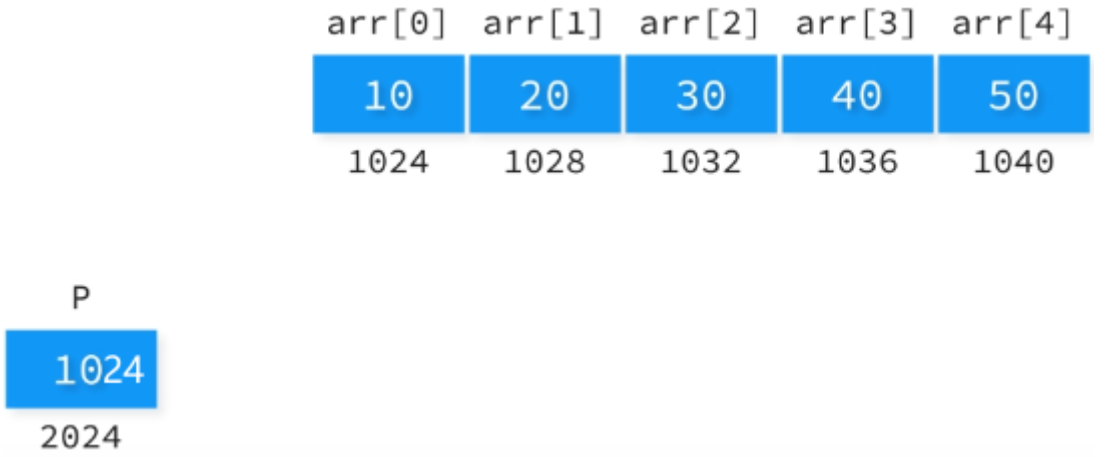
```
ptr   = 1024
ptr+1 = 1028
ptr+3 = 1036
```

```
int arr[5] = {10, 20, 30, 40, 50};
int *P;

P = NULL;

if(P == NULL)
    printf("P is NULL\n");
```




```
int arr[5] = {10, 20, 30, 40, 50};
int *P;

P = &arr[0];

if(P != NULL)
    printf("arr[0] = %d\n",*P);
```

p 的地址是 arr[0]的地址。



```
int *ptr;

ptr = malloc(5 * sizeof(int));

*(ptr+0) = 10;
*(ptr+1) = 20;
*(ptr+2) = 30;
*(ptr+3) = 40;
*(ptr+4) = 50;
```