

## pointer-to-pointer (double pointer)

- "pass by pointer" of function parameter
  - means passing a pointer by value
  - In most cases, no problem!
  - Problem comes when modifying the pointer inside the function
    - In this case, we need pointer-to-pointer.

```
int g_One=1;
void func(int* pInt);

int main()
{
    int nvar=2;
    int* pvar=&nvar;
    func(pvar);
    printf("%d",*pvar); // output???

    return 0;
}

void func(int* pInt)
{
    pInt=&g_One;
}
```

## Pointer-to-Pointer parameter

```
void func(int** ppInt);
int g_One=5;

int main()
{
    int nvar=2;
    int* pvar=&nvar;
    func(&pvar); // this is necessary to modify the pointer pvar in the function
    printf("%d",*pvar); // output???
    return 0;
}

void func(int** ppInt) // ppInt takes the address of pointer variable
{
    //Modify the pointer, ppInt points to
    *ppInt=&g_One;

    //You can also allocate memory, depending on your requirements
    /**ppInt=(int*)malloc(sizeof(int));
    /**ppInt=10;

    //Modify the variable, *ppInt points to
    /**ppInt=3;
}
```

## example

- Is this OK???

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

int main()
{
    int* p, i;
    func(p);
    for (i=0;i<5;i++) scanf("%d",&p[i]);
    for (i=0;i<5;i++) printf("p[%d]=%d\n",i,p[i]);
    return 0;
}

void func(int* ptr)
{
    ptr=(int*)malloc(5*sizeof(int));
}
```