APS 105 Lecture 13 Notes

Last lecture: Two ways to communicate into functions: Using "call by value" or pointers.

Today: More on pointers (passing pointers to functions), double pointers and "returning" a pointer

Recall swap function:

We want to implement a function the swaps the values of 2 integers.

Do we need pointers? Yes.

Why? If we swap two integers in a function where parameters were passed by value, the swapping will happen only in the scope of the function.
To have the integers swapped in the main function, we need to provide the function with the addresses of the 2 integers to swap.

IN DEMO, implement swap function with & without pointers to show the difference!

Correct swap

```
void swap(int *px, int *py){

    int temp = *p;
    *p = *q;
    *q = temp;
}

int main (void){

    int a=1, b=2;
    swap(&a, &b);
    printf("a is %d, b is %d", a, b);
    return 0;
}
```

address of a is passed to px, address of b is passed to py

More exercises on pointers:

If I have the following code:

```
int i;
int * pi;
double d;
double * pd;
```

Which is a valid statement(s)?

```
i = &pd;        i is int    &pd is address of pd
pi = &i;
pd = i;
pd = &pi;
*pi = &i;
*pd = 7.0;
```

What is the size of the pointer varidble?

It depends on how do I represent an address. Old machines used 32-bit address. Modern machines use 64-bit address.

```
int *pi;
double *pd;
printf("Size of pi is %d, size of pd is %d", sizeof(pi),
                                             sizeof(pd));
```

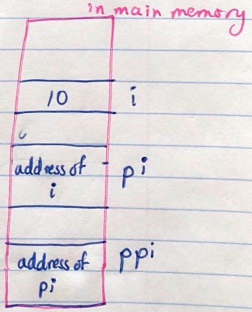
Can a pointer hold the address of another pointer?

Yes, but its type with be double pointer **.

E.g.

```
int i;
int *pi;
int **ppi;

i= 10;
pi = &i;
ppi = &pi;
```



Can I declare a pointer & initialize it in the same statement?

Yes

If I can do `int i = 10;`

I can also do `int *pi = &i;` $\xrightarrow{\text{equivalent to}}$ `int *pi;`
`pi = &i;`

Can a function return a pointer?

E.g Implement a function that returns a pointer to the location with largest value.

Function input is 2 pointers to double
Function return/output is 1 pointer to the largest double

The function returns a pointer to double

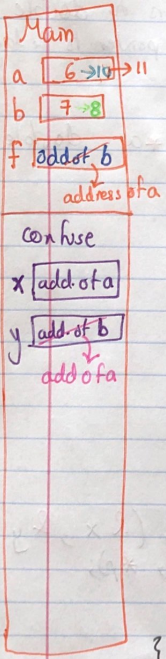
```
double *largeValLoc ( double *a , double *b){
        if ( *a > *b )
                return a;
        else
                return b;
}

int main (void){

        double x = 2.6, y = 7.3;
        double *p = largeValLoc (&x , &y );
        printf(" Large value is %lf", *p);

}
```

Practice Question: Fall'18 Midterm Q7



```
#include <stdio.h>

int * confuse (int * x, int *y){
       b
  ① (*y) ++;     b++
  ② y = x;       y is now pointing to a
  ③ *y = 10;     a = 10;
    a
  ④ return y;    confuse function returns a pointer to a

}

int main (void){

   int a = 6, b = 7;
   int *f = &b;
                  address of a is returned
   f = confuse( &a, &b);
   (*f) ++;   a++
    a
   printf("a= %d  and b = %d", a, b);
               a = 11 and b = 8
   return 0;

}
```