Pointer review

draw memory!

```
void foo(int *pArg, int val)
{
   int x = 10;
   int y;

   val = 300;
   *pArg = x;
   pArg = &y;
  *pArg = val;
   pArg = (int*) malloc(sizeof(int));
  *pArg = 99;
}

int main()
{
  int other = 7;
  int num = 9;
  int *pInt = & num;
  int *pVal = pInt;

  pInt = (int*) malloc(sizeof(int));
  *pInt = 250;

   foo(pVal, other);

   free(pInt);
}
```

Continue drawing memory!

C Operator Precedence

++ --

[] Array subscripting

. Struct member access

->

All of the above have a higher precedence than indirection/dereference *

https://en.cppreference.com/w/c/language/operator_precedence

Handles, a pointer to a pointer

```
int **hInt;
int *pInt;
int actualInt = 2;
int otherInt = 4
// draw the memory!

pInt = &actualInt;
hInt = &pInt;

pInt =

*pInt =

hInt =

*hInt =

**hInt =

*hInt = & otherInt;

pInt =
```

WHY!?!?

Create

void createArray(int **hArray, int size, int fill)

void resizeArray(int **hArray, int oldSize, int newSize, int fill)

Practice

Node sList;

NodePtr psList;

NodePtr *hsList;

```c
typedef struct Node *NodePtr;
typedef struct Node
{
  char data;
  NodePtr psNext;
} Node;
```

```c
// create an empty List
void lstCreate(NodePtr *hsList);

void lstInsertFirst(NodePtr *hsList, char data);

void lstDeleteFirst(NodePtr *hsList, char* pData);

void lstTerminate(NodePtr *hsList);
```

```
NodePtr psList;
int i;

lstCreate(&psList);

for(i = 0 ; i < 10 ; i++)
{
  lstInsertFirst(&psList, 'a' + i);
}
```