

Exercise Zero

```
[scole4@gaea lab3]$ git clone https://vcs.cs.unb.ca/git/cs2263-scol
Cloning into 'cs2263-scole4'...
Username for 'https://vcs.cs.unb.ca': scole4
Password for 'https://scole4@vcs.cs.unb.ca':
warning: You appear to have cloned an empty repository.
[scole4@gaea lab3]$ ls
cs2263-scole4
[scole4@gaea lab3]$ cd cs2263-scole4/
```

Exercise One

arithmetic.c

```
#include <stdio.h>
#include <stdlib.h>
int main (int argc ,char * * argv)
{
    int  arr1[] = {7, 2, 5, 3, 1, 6, -8, 16, 4};
    char arr2[] = {'m', 'q', 'k', 'z', '%', '>'};
    double arr3[] = {3.14, -2.718, 6.626, 0.529};

    int len1 = sizeof(arr1) / sizeof(int);
    int len2 = sizeof(arr2) / sizeof(char);
    int len3 = sizeof(arr3) / sizeof(double);
    printf("lengths = %d, %d, %d\n", len1, len2, len3);

    int  * iptr = arr1;
    char  * cptr = arr2;
    double * dptr = arr3;
    printf("values = %p, %p, %p\n", iptr, cptr, dptr);

    iptr ++;
    cptr ++;
    dptr ++;
    printf("values = %p, %p, %p\n", iptr, cptr, dptr);
```

```

iptr ++;
cptr ++;
dptr ++;
printf("values = %p, %p, %p\n", iptr, cptr, dptr);

iptr ++;
cptr ++;
dptr ++;
printf("values = %p, %p, %p\n", iptr, cptr, dptr);

return EXIT_SUCCESS;
}

```

```

[scole4@gaea cs2263-scole4]$ gcc -o test -Wall arithmetic1.c
[scole4@gaea cs2263-scole4]$ ./test
lengths = 9, 6, 4
values = 0x7ffddc41ae20, 0x7ffddc41ae10, 0x7ffddc41adf0
values = 0x7ffddc41ae24, 0x7ffddc41ae11, 0x7ffddc41adf8
values = 0x7ffddc41ae28, 0x7ffddc41ae12, 0x7ffddc41ae00
values = 0x7ffddc41ae2c, 0x7ffddc41ae13, 0x7ffddc41ae08

```

Are the pointer variables incremented correctly? Show your calculations based on the memory addresses printed by your program.

Yes, they are. Integer size = 4 Bytes, Character = 1 Byte and Double = 8 Byte.

iptr: $0x7ffddc41ae20 + 4 = 0x7ffddc41ae24 + 4 = 0x7ffddc41ae28$

cptr: $0x7ffddc41ae10 + 1 = 0x7ffddc41ae11 + 1 = 0x7ffddc41ae12$

dptr: $0x7ffddc41adf0 + 8 = 0x7ffddc41adf8 + 8 = 0x7ffddc41ae00$

Are the increments for different pointers the same? Explain why.

The increments for pointers differ based on their data type. Integer size = 4 Bytes, Character = 1 Byte and Double = 8 Byte. Based on the type of the array they will increment by the size of the data type they contain.

Exercise Two

loopbyaddress.c

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

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

```
int main(void)
{
    int arr[] = {10, 11, 12, 13, 14, 15, 16};
    int size = 7;
    int* anchor = &arr[0];
    int i;
    int* next = anchor;
    int loops = 0;

    for(i=0; i<=size; i++)
    {
        if(loops == 2)
            break;
        else if(i == size)
        {
            next = anchor;
            i=-1;
            loops++;
            continue;
        }

        printf("%d %d %p %d\n", i, *next, next, *next);
        next++;
    }
    return 0;
}
```

```
[scole4@gaea cs2263-scole4]$ ./test
0 10 0x7fff2b2b8650 10
1 11 0x7fff2b2b8654 11
2 12 0x7fff2b2b8658 12
3 13 0x7fff2b2b865c 13
4 14 0x7fff2b2b8660 14
5 15 0x7fff2b2b8664 15
6 16 0x7fff2b2b8668 16
0 10 0x7fff2b2b8650 10
1 11 0x7fff2b2b8654 11
2 12 0x7fff2b2b8658 12
3 13 0x7fff2b2b865c 13
4 14 0x7fff2b2b8660 14
5 15 0x7fff2b2b8664 15
6 16 0x7fff2b2b8668 16
```

Push for Exercise One and Exercise Two

```

[scole4@gaea cs2263-scole4]$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       arithmetic1.c
#       loopbyaddress.c
nothing added to commit but untracked files present (use "git add"
[scole4@gaea cs2263-scole4]$ git add .
[scole4@gaea cs2263-scole4]$ git commit -m "lab3"
[master (root-commit) 9a69f34] lab3
 2 files changed, 64 insertions(+)
 create mode 100644 arithmetic1.c
 create mode 100644 loopbyaddress.c
[scole4@gaea cs2263-scole4]$ git push origin master
Username for 'https://vcs.cs.unb.ca': scole4
Password for 'https://scole4@vcs.cs.unb.ca':
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 819 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://vcs.cs.unb.ca/git/cs2263-scole4
 * [new branch]      master -> master

```

Exercise Three

arrindex.c

```

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

```

```

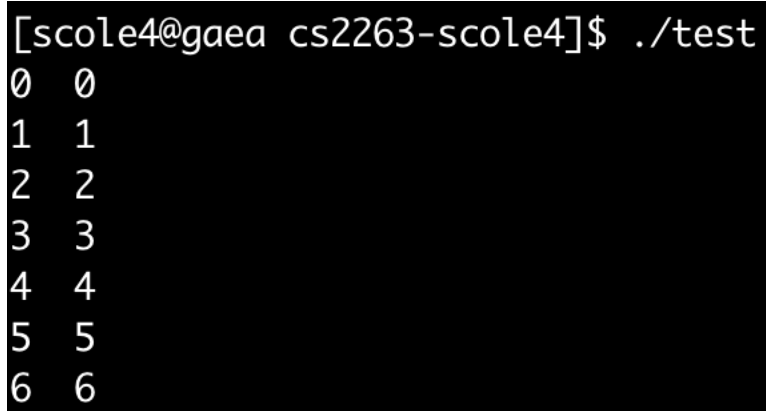
int main(void)
{
    int arr[] = {10, 11, 12, 13, 14, 15, 16};
    int i;
    for (i=0; i< sizeof(arr)/sizeof(arr[0]); i++)
        printf("%d %d \n", i, arrindex( &arr[0], &arr[i]));
}

int arrindex(int * p1, int * p2)
{
    int* address = p2;
    int i;

    for(i=0; address >= p1; i++)
    {
        if(address == p1)
            return i;
        address--;
    }

    return 0;
}

```



```

[scole4@gaea cs2263-scole4]$ ./test
0 0
1 1
2 2
3 3
4 4
5 5
6 6

```

Exercise Four

wrongindex.c

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char * * argv)
{
    int x = -2;

```

```

int arr[] = {0, 1, 2, 3, 4};
int y = 15;
printf("& x      = %p, & y      = %p\n", & x, & y);
printf("& arr[0] = %p, & arr[4] = %p\n", & arr[0],
& arr[4]);
printf("x = %d, y = %d\n", x, y);
arr[-1] = 7;
arr[5] = -23;
printf("x = %d, y = %d\n", x, y);
arr[6] = 108;
printf("x = %d, y = %d\n", x, y);
arr[7] = -353;
printf("x = %d, y = %d\n", x, y);

printf("x %p, y %p\n", &x, &y);

int i;
for(i=-1; i<7; i++)
    printf("Index %d: %p\n", i, &arr[i]);

return EXIT_SUCCESS;
}
[scole4@gaea cs2263-scole4]$ ./test
& x      = 0x7ffd2fad6878, & y      = 0x7ffd2fad685c
& arr[0] = 0x7ffd2fad6860, & arr[4] = 0x7ffd2fad6870
x = -2, y = 15
x = -2, y = 7
x = 108, y = 7
x = 108, y = 7
x 0x7ffd2fad6878, y 0x7ffd2fad685c
Index -1: 0x7ffd2fad685c
Index 0: 0x7ffd2fad6860
Index 1: 0x7ffd2fad6864
Index 2: 0x7ffd2fad6868
Index 3: 0x7ffd2fad686c
Index 4: 0x7ffd2fad6870
Index 5: 0x7ffd2fad6874
Index 6: 0x7ffd2fad6878

```

The results from the textbook are different because the array is indexed in a slightly different location. In my example the location of `arr[-1] == y`; therefore, `y` is set to the value intended for `arr[-1]` and the same for the address of `arr[6]` and `x`.

Push for Exercise Three and Four

```
[scole4@gaea cs2263-scole4]$ git push origin master
Username for 'https://vcs.cs.unb.ca': scole4
Password for 'https://scole4@vcs.cs.unb.ca':
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 502 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://vcs.cs.unb.ca/git/cs2263-scole4
614de5d..a1919ee master -> master
```

Exercise Five

```
[scole4@gaea cs2263-scole4]$ ls
arithmetic1.c  arrindex.c  loopbyaddress.c  test  wrongindex.c  wronginde
[scole4@gaea cs2263-scole4]$ rm test
[scole4@gaea cs2263-scole4]$ ls
arithmetic1.c  arrindex.c  loopbyaddress.c  wrongindex.c  wronginde
[scole4@gaea cs2263-scole4]$ git checkout test
[scole4@gaea cs2263-scole4]$ ls
arithmetic1.c  arrindex.c  loopbyaddress.c  test  wrongindex.c  wronginde
5  1  40  2263  1  47$
```