COMP1911 22T2 (https://webcms3.cse.unsw.edu.au/COMP1911/22T2) **Code Examples from Lectures on 12-8_dynamicMemory** Introduction to Programming (https://webcms3.cse.unsw.edu.au/COMP1911/22T2)

banana.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/banana.c)

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

int *createArray() {
    int *array = malloc(sizeof(int) * 5);
    return array;
}

int main(int argc, char* argv[]) {
    int *array = createArray();
    int i = 0;
    while (i < 5) {
        array[i] = i;
        i++;
    }
    i = 0;
    while (i < 5) {
        printf("array[%d] = %d\n", i, array[i]);
        i++;
    }
    free(array);
    array = NULL;
    return 0;
}
```

char.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/char.c)

```
#include <stdio.h>

int main(int argc, char* argv[]) {
    int hundreds = getchar() - 48;
    int tens = getchar() - 48;
    int ones = getchar() - 48;

    int number = hundreds*100 + tens*10 + ones;

    printf("%d %d %d = %d", hundreds, tens, ones, number * 2);
    putchar('\n');
    return 0;
}
```

crap.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/crap.c)

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

int createInt() {
    int i = 5;
    return i;
}

int main(int argc, char* argv[]) {
    int a = createInt();
    int i = 0;
    return 0;
}
```

mallocfree.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/mallocfree.c)

manage.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/manage.c)

```
#include <stdio.h>

void stupidFunction() {
    int stupid = 10;
    printf("%d\n", stupid);
}

int main(int argc, char* argv[]) {
    int apple = 3;
    int banana = 4;
    int i = 0;
    while (i < 10000) {
        stupidFunction();
        i++;
    }

    return 0;
}
```

manage2.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/manage2.c)

```
#include <stdio.h>

void stupidFunction() {
    int stupid = 10;
    printf("%d\n", stupid);
}

int main(int argc, char* argv[]) {
    int apple = 3;
    int banana = 4;
    int i = 0;
    while (i < 10000) {
        stupidFunction();
        i++;
    }

    return 0;
}
```

test.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/test.c)

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

void create() {
    int *s = malloc(sizeof(int));
}

int main(int argc, char* argv[]) {
    create();
    return 0;
}
```

arg.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/arg.c)

```
#include <stdio.h>

int main(int argc, char* argv[]) {

    char *s = "abcdef";
    printf("Third character: %c\n", s[2]);

    int i = 1;
    while (i < argc) {
        printf("Argument %d = %s\n", i, argv[i]);
        i++;
    }
    printf("Magical character: %c\n", argv[2][0]);
    return 0;
}
```

2dmalloc.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/12-8_dynamicMemory/code/2dmalloc.c)

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

#define SIZE 4

1. A B C D
2. E F G H
3. I J K L
4. M N O P

int main(int argc, char* argv[]) {
    int **a = malloc(size(int*) * SIZE);
    int i = 0;
    while (i < SIZE) {
        a[i] = malloc(size(int) * SIZE);
        i++;
    }

    return 0;
}

/*

a
[0] => [A, B, C, D]
[1] => [E, F, G, H]
[2] => [I, J, K, L]
[3] => [M, N, O, P]

*/
```