```c
//
//  main.c
//  Array
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates arrays.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    // prompt user for number of exams
    int n;
    printf("Enter number of exams: ");
    scanf("%d", &n);

    // allocate memory for grades (on stack)
    int grades[n];

    // prompt user for exams' grades
    for (int i = 0; i < n; i++) {
        printf("Enter grade %d of %d: ", i+1, n);
        scanf("%d", &grades[i]);
    }

    // do something with grades...

    return 0;
}
```

```c
//
//  main.c
//  Cast
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates explicit casting between primitives.
//

#include <stdio.h>

int main (int argc, const char * argv[])
{
    char c = 'A';
    int i = (int) c;
    printf("%c = %d\n", c, i);
    return 0;
}
```

```c
//
//  main.c
//  Conditions
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Reports whether user's input is positive, negative, or zero.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    int n;
    printf("Enter an integer: ");
    scanf("%d", &n);
    if (n > 0) {
        printf("Thanks for the positive integer!\n");
    }
    else if (n < 0) {
        printf("Thanks for the negative integer!\n");
    }
    else {
        printf("Thanks for the zero!\n");
    }
    return 0;
}
```

```c
//
//  main.c
//  DoWhile
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates a do-while loop.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    int n;
    do {
        printf("Enter a positive integer: ");
        scanf("%d", &n);
    }
    while (n < 1);
    printf("Thanks for the positive integer!\n");
    return 0;
}
```

```c
//
//  main.c
//  Enum
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates enum (and struct).
//

#include <stdio.h>

// gender
typedef enum {
    FEMALE,
    MALE
} genders;

// student
typedef struct {
    char *name;
    genders gender;
} student;

// prototype
void greet(student s);

int main (int argc, const char * argv[])
{
    // alice
    student alice;
    alice.name = "Alice";
    alice.gender = FEMALE;
    greet(alice);

    // bob
    student bob;
    bob.name = "Bob";
    bob.gender = MALE;
    greet(bob);

    return 0;
}

// greets student
void greet(student s)
{
```

```c
    char *title = (s.gender == FEMALE) ? "Ms." : "Mr.";
    printf("Hello, %s %s.\n", title, s.name);
}
```

```c
//
//  main.c
//  For
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates a for loop.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for (int i = n; i > 0; i--) {
        printf("%d...\n", i);
    }
    printf("Blast off!\n");
    return 0;
}
```

```c
//
//  main.c
//  GetInt
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Gets an int from the user.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    int n;
    printf("Enter an integer: ");
    scanf("%d", &n);
    printf("Thanks for the %d!\n", n);
    return 0;
}
```

```c
//
//  main.c
//  HelloC
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Says hello to the world in C.
//

#include <stdio.h>

int main (int argc, const char * argv[])
{
    printf("Hello, World!\n");
    return 0;
}
```

```
//
// Prefix header for all source files of the 'HelloObjC' target in the 'HelloObjC' project
//

#ifdef __OBJC__
    #import <Foundation/Foundation.h>
#endif
```

```
1. //
2. //  main.c
3. //  HelloObjC
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Says hello to the world in Objective-C.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. int main (int argc, const char * argv[])
15. {
16.     NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
17.     NSLog(@"Hello, World!");
18.     [pool drain];
19.     return 0;
20. }
```

```c
//
//  main.c
//  Malloc
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates malloc.
//

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

// prototype
int *get_grades(int exams);

int main (int argc, const char * argv[])
{
    // prompt user for number of exams
    int n;
    printf("Enter number of exams: ");
    scanf("%d", &n);

    // get grades
    int *grades = get_grades(n);

    // do something with grades...

    // free memory
    free(grades);

    return 0;
}

// gets grades
int *get_grades(int exams)
{
    // allocate memory for grades (on heap)
    int *grades = malloc(sizeof(int) * exams);

    // prompt user for exams' grades
    for (int i = 0; i < exams; i++) {
        printf("Enter grade %d of %d: ", i+1, exams);
        scanf("%d", &grades[i]);
    }
    return grades;
}
```

```c
//
//  main.c
//  SizeOfPrimitives
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Prints some pointers' sizes.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    printf("double: %lu\n", sizeof(double *));
    printf("float: %lu\n", sizeof(float *));
    printf("int: %lu\n", sizeof(int *));
    printf("long long: %lu\n", sizeof(long long *));
    return 0;
}
```

```c
//
//  main.c
//  SizeOfPrimitives
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Prints some primitives' sizes.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    printf("double: %lu\n", sizeof(double));
    printf("float: %lu\n", sizeof(float));
    printf("int: %lu\n", sizeof(int));
    printf("long long: %lu\n", sizeof(long long));
    return 0;
}
```

```c
//
//  main.c
//  Struct
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates struct.
//

#include <stdio.h>

// student
typedef struct {
    int age;
    char *name;
} student;

// prototype
void greet(student s);

int main (int argc, const char * argv[])
{
    // alice
    student alice;
    alice.age = 20;
    alice.name = "Alice";
    greet(alice);

    // bob
    student bob;
    bob.age = 21;
    bob.name = "Bob";
    greet(bob);

    return 0;
}

// greets student
void greet(student s)
{
    printf("Hello, %s.  I see that you are %d years old.\n", s.name, s.age);
}
```

```
1. //
2. //  Student.h
3. //  Students1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14.
15. @interface Student : NSObject {
16. @public
17.     int age;
18.     NSString *name;
19. }
20. @end
```

```
1.  //
2.  //  Student.m
3.  //  Students1
4.  //
5.  //  David J. Malan
6.  //  Harvard University
7.  //  dmalan@harvard.edu
8.  //
9.  //  Defines a student.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15. @end
```

```
//
// Prefix header for all source files of the 'Students1' target in the 'Students1' project
//

#ifdef __OBJC__
    #import <Foundation/Foundation.h>
#endif
```

```
//
//  main.m
//  Students1
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates use of a class.
//

#import <Foundation/Foundation.h>
#import "Student.h"

// prototype
void greet(Student *s);

int main (int argc, const char * argv[])
{
    // allocate autorelease pool
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];

    // Alice
    Student *alice = [Student alloc];
    alice->age = 20;
    alice->name = @"Alice";
    greet(alice);
    [alice release];

    // Bob
    Student *bob = [Student alloc];
    bob->age = 21;
    bob->name = @"Bob";
    greet(bob);
    [bob release];

    // drain autorelease pool
    [pool drain];

    return 0;
}

// greets student (via stderr)
void greet(Student *s)
{
    NSLog(@"Hello, %@.  I see that are %d years old.\n", s->name, s->age);
}
```

```c
//
//  main.c
//  SwapFailure
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Fails to swap two variables' values.
//

#include <stdio.h>

// function prototype
void swap(int a, int b);


int main(int argc, const char * argv[])
{
    int x = 0;
    int y = 1;

    printf("x is %d\n", x);
    printf("y is %d\n", y);
    printf("Swapping x and y...\n");
    swap(x, y);
    printf("Success!\n");
    printf("x is %d\n", x);
    printf("y is %d\n", y);

    return 0;
}


//
// Swaps arguments' values.
//

void swap(int a, int b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
```

```c
//
//  main.c
//  SwapSuccess
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Swaps two variables' values.
//

#include <stdio.h>

// function prototype
void swap(int *a, int *b);


int main(int argc, const char * argv[])
{
    int x = 0;
    int y = 1;

    printf("x is %d\n", x);
    printf("y is %d\n", y);
    printf("Swapping x and y...\n");
    swap(&x, &y);
    printf("Success!\n");
    printf("x is %d\n", x);
    printf("y is %d\n", y);

    return 0;
}


//
// Swaps arguments' values.
//

void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

```c
//
//  main.c
//  While
//
//  David J. Malan
//  Harvard University
//  dmalan@harvard.edu
//
//  Demonstrates a while loop.
//

#include <stdio.h>

int main(int argc, const char * argv[])
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    while (n > 0) {
        printf("%d...\n", n);
        n--;
    }
    printf("Blast off!\n");
    return 0;
}
```