

## B2 • Discussion #13 • 2021-04-23

GOOD LUCK ON THE EXAM!

### Announcements

- Exam 3 is today from 4:30-6:00, with 15 more minutes for uploading
  - Be careful on the exam
  - You may use C, but it is strongly recommended that you do not, as it is likely that doing so will cost you time. The exam is about your knowledge, not what you can look up.
  - Programs must follow good programming style but you don't have to write comments unless specified by the problem
  - DON'T MAKE THINGS HARDER THAN THEY ARE!!
  - Write GENERAL code
  - Topics are ALL of the C topics covered
- Next Monday: in the morning, time for you to do the Course Evals, and there will be Open Hours. Lab attendance is required, it will be optional project presentations.
- Final Project due next Wednesday by 10am for everyone. Upload one document per group to Gradescope. PLEASE make sure all names are on it.

Questions about the material?

Drop 'em in the chat during announcements



### Quiz #8 Review

## Typecasting: (type) expression

ex.

```
float number;  
number = 1/5; ← 0.00000 ← integer division: 1/5 will give 0, then bc  
number = (float) 1/5; ← 0.2 ← number is a float, it's 0.0  
                        (float) 1/5 means 1.0/5
```

## Generating random integers

- time.h
- stdlib.h

```
rand(time(NULL));
```

general formula:

$$\text{rand()} \% (\text{MAX} - \text{MIN} + 1) + \text{MIN}$$

ex. generate from 5 to 10

$$\text{rand()} \% (10 - 5 + 1) + 5$$
$$\text{rand()} \% 6 + 5$$

generating random float:

$$(\text{float}) \text{rand()} / N$$

## Shortcuts

count++ vs. ++count

ex.

```
int num = 5;
```

```
printf("The num is %d\n", num--);
```

// The num is 5 (now num is 4)

```
printf("The num is %d\n", --num); ← now num is 3
```

// The num is 3

```
++ ____      ____ += 2 ← increment by 2
```

```
____ ++
```

```
-- ____
```

```
____ --
```

```
var + 2
```

```
#define N 6
```

```
for (i = 0; i < N; i += 2)  
{  
    printf("i is %d\n", i);  
}
```

```
printf("i is %d\n", i);
```

i is 0

i is 2

i is 4

i is 6

← not part of the loop

same thing as  
 $i = i + 2$

## draw\_triangle.c

void draw\_me(char, int); ← prototype

draw\_me(symbol, side); ← w/in int main()

```
void draw_me(char sym, int side_len)
{
    int i, j;
```

```
    for (i = 1; i <= side_len; i++) ← you can start @ 0 and then do i <= side_len
    {
```

```
        for (j = 1; j <= side_len; j++)
```

```
        {
            printf("%.3c", sym);
```

```
        }
        printf("\n");
```

```
    }
```

field width: sym = !

%.3c

— — !

%. -3c

! — —

## Double Pointers

void myDMAfunction(int \*\*, char \*\*); ← prototype

myDMAfunction(&intptr, &chptr); ← function call w/in int main()

void myDMAfunction(int \*\*ipt, char \*\*chp) ← function definition

```
{
    *ipt = (int *) malloc(sizeof(int));
    *chp = (char *) malloc(sizeof(char));
```

```
    **ipt = 30;
```

```
    **chp = 'c';
```

```
}
```

## Typedef of struct

```
#define N 5
```

```
typedef struct  
{  
    int num;  
    char word[10];  
} mystruct;
```

charlesstruct mystruct

num	word
int	char[]

```
int main()  
{  
    mystruct charlesstruct[N];  
    int i;  
    char myword[10]; ← ----- 10
```

```
    for (i = 0; i < N; i++)  
    {  
        printf("Enter a word: ");  
        scanf("%s", myword);  
        strcpy(charlesstruct[i].word, myword);  
        fflush(stdin);  
    }
```

```
    for (i = 0; i < N; i++)  
    {  
        printf("i is %d and the word is %s\n", i, charlesstruct[i].word);  
        printf("The %dth character is %c\n", i, charlesstruct[i].word[i]);  
    }  
    return 0;  
}
```