

BE Discussion 10, 2020-11-06

Announcements

- today is last day to drop a class @ BU
- Exam 2 median 90.5
- Next week don't need to use gcc compiler
- this week and next, reinforcing concepts already learned in MATLAB, new material after that
- HW #4 due Monday at 9am EST
- Final project + proposal also released
- Quiz this afternoon 4:40 pm EST - 5:00pm, 15 min to complete, 5 min to upload
 - check formatting on Gradescope
 - Alternate time zone @ 7:00 am EST Saturday
- check the syllabus!

Review of Material

Header Files

- <stdio.h> ← needed for i/o
- <stdlib.h> ← needed for random number generation
- <time.h> ← needed for strings
- <math.h> ← needed for math operations

Typecasting: (type) expression

- float number;
- number = 1/5; ← 0.0000
- number = (float) 1/5; ← 0.2000

Generating Random Integers:

- 2 header files:
 - <stdlib.h>
 - <time.h>
- in C, you always need to set the seed
srand(time(NULL)); * inside int main() { }

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

ex. generate in range from 0 to N
 $\text{rand()} \% (N+1)$

0 to 10
← $\text{rand()} \% 11$

MAX: 10
MIN: 0
 $(10 - 0 + 1) + 0$
% 11

ex. generate in range from 1 to N
 $\text{rand()} \% N + 1$

1 to 10
← $\text{rand()} \% 10 + 1$

MAX: 10
MIN: 1
 $(10 - 1 + 1) + 1$
% 10 + 1

ex. generate in range from K to N
 $\text{rand()} \% (N - K + 1) + K$

↑ if we did $\text{rand()} \% 10$: 0 to 9
→ add 1
1 to 10

Shortcuts

count++ v.s. ++count

subtraction happens
after, so decrement
after print

```
int num = 5;  
printf("The num is %d\n", num--);  
// The num is 5  
printf("The num is %d\n", --num);  
// The num is 3  
// decrements before so 3 is printed
```

ex. generate in range from 5 to 10

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

$\text{rand()} \% 6 + 5$ ← first, generating from 0 to 5, second adding 5

ex. generate a random float

$(\text{float}) \text{rand()} / N$ ← range is 0 to N (don't worry about this)

Flushing the input buffer

flush(stdin); fflush(stdin);

Selection Statements

```
if (condition expression)
{
    // general statement
    // other statement
}
else
{
    // different statement
}
```

Switch Example

```
switch (variable)
{
    case 0:
    case 1:
        printf("For case 0 or case 1\n");
        break;
    case 2:
        printf("For case 2\n");
        break;
    default
        printf("The default.\n");
}
```

Loops !!!

- for loop:

```
for (i = 0; i <= N; i++)  
    printf("Hi!\n")
```

- Nested for loop:

```
for (i = 0; i <= N; i++)  
{  
    //action in outer loop, including inner  
    for (j = 0; j <= K; j++)  
    {  
        //action in inner loop  
    }  
}
```

- While loop: action may be skipped entirely

```
while (condition expression)  
{  
    action_1;  
    action_2;  
}
```

- do while loop: action on top, therefore always executed at least once!

```
do  
{  
    action_1;  
    action_2;  
}  
while (condition expression);
```

DO NOT WORRY ABOUT THIS MATERIAL, BUT SINCE IT WAS ASKED:

Assume that the file quizfile.dat contains exactly the following:

```
c 33
d 44
c 1
a 22
```

```
#include <stdio.h>
int main()
{
    int value,
        sum = 0,
        i,
        j;
    char code;
    FILE *ifp;
    ifp = fopen("quizfile.dat", "r");

    while(fscanf(ifp, "%c%d\n", &code, &value) != EOF)
    {
        if (code == 'c')
            sum += value;
    }
    printf("The sum is %d.\n", sum);

    for(i=5; i <= 8; i+=2)
        printf("i is %d, ", i);
    printf("\nand now i is %d\n", i);

    for(i=1; i <= 3; i++)
    {
        printf("Round %d: ", i);
        for(j=5; j >= i; j--)
            printf("*");
        printf("\n");
    }
    fclose(ifp);
    return 0;
}
```

we call this a pointer,
don't worry about this
right now

program output:

The sum is 34

i is 5, i is 7,

and now i is 9

Round 1: * * * * *

Round 2: * * * *

Round 3: * * *