CSC180 tutorial notes – week of Oct. 27

Review of ++ and –

This:
```c
int n = 10;
printf("%d", n++);
printf("%d", n);
```
prints 10,11
But this:
```c
int n = 10;
printf("%d", ++n);
printf("%d", n);
```
prints 11,11

-- works the same way

The getline() function in A2

We discuss this function from the assignment:

```c
int getline(char s[], int lim)
{
    int c, i;
    i = 0;
    while (--lim > 0 && (c=getchar()) != EOF && c != '\n')
        s[i++] = c;
    if (c == '\n')
        s[i++] = c;
    s[i] = '\0';
    return i;
}
```

The input for A2 is a sequence of characters terminated by EOF. There are two ways to provide input:

- ./receipt < input.txt. EOF can be considered the last character in input.txt
- Type in the input using a keyboard, use cntrl-d in Linux to indicate EOF(suggest not doing that, and it’s different in Windows)
- lim is the size of the array s. If the lines are longer than s-1, there’s a problem.
- If all the lines in your input are smaller than lim in size(and you get to pick lim), you can call getline() multiple times, and obtain all the lines in your input sequentially.
- getline() helps you by reading the input line-by-line. You can then analyze the lines to get the information you need.
- getline() doesn’t put ‘\n’ at the end of s if and only if we encountered EOF.
Using sscanf()

sscanf() can be used to analyze strings. Example: if we know that the string str is of the form “my name is X”, we can do this:

```c
int age;
char name[200];
char str[] = "My name is John and I am 18 years old";
sscanf(str, "My name is %s and I am %d years old", name, &age);
```

If the contents don’t match in the format string and the input string, sscanf returns. Note: scanf doesn’t work this way generally.

ROSI:

We show how to use a double array as a table for grades:
int n_names = 4;
int n_courses = 4;
char *names[] = {"John", "Jill", "Jack", "Jane");
char *courses[] = {"Calculus", "Algebra", "Mechanics", "Programming"};
int grades[4][4];

int get_index(char *str, char **items, int n_items)
{
    int i;
    for (i = 0; i < n_items; i++)
    {
        if (strcmp(str, items[i]) == 0)
        {
            return i;
        }
    }
    return -1;
}

void accept_bribe()
{
    int i, index_name;
    char name[200];
    printf("Enter name:\n");
    scanf("%s", name);
    index_name = get_index(name, names, n_names);
    if (index_name == -1)
    {
        return;
    }
    for (i = 0; i < n_courses; i++)
    {
        grades[index_name][i] = 100;
    }
}

void enter_grade()
{
    int index_name, index_course;
    char name[200];
    char course[200];
    printf("Enter name:\n");
    scanf("%s", name);
    printf("Enter course:\n");
    scanf("%s", course);
    index_name = get_index(name, names, n_names);
    index_course = get_index(course, courses, n_names);
    if ( (index_name == -1) || (index_course == -1) )
    {
        return;
    }
    printf("Enter grade: \n");
    scanf("%d", &grades[index_name][index_course]);
}