COMP10002 Workshop Week 7

Outlook:

1	Strings and some problems from lec06.pdf: Discuss the role of arrays and strings as presented in the lec06.pdf lectures slides, and looking at the first three Exercises in those slides (functions is_subsequence(), is_subset(), and is_anagram()). Implement and test at least one of those three functions.
2	Assignment 1: understanding, requirements, submission
3	Lab: Working on assignment 1

A string `s` is an array of `strlen(s)` elements

A string s is a sequence of chars starting from s and ending at the first occurrence of the sentinel '\0'

Strings: traversal – method 1

Suppose that `s` is a string. Traversal of `s` means "visiting" each of the character of s exactly once. "visit" mean "do some job with".

Method 1:

```
int i;
for  (i=0; i<strlen(s); i++){
     visit s[i] ;
}
```

Method 2:

```
char *p;
for (p=s; *p ; p++) {
    visit *p ;  // *p is s[i]
}
```

Strings: traversal

Example:

count frequencies of alphabetic characters in a string, *ignoring case*. Start with:

```
void count_freq(char *s, int freq[]) {
    int i; char *p;
    for(i=0; i<N; i++) freq[i]= 0;




}
```

Method 1? Method 2?

String input: method 1= scanf (simple, not working sometimes)

Given the declaration:

```
char s[MAXCHARS+1];
```

How to get `s` from standard input, supposing that '`\n`' is the end of string? For example, the input:

```
atgatccccg
The cat in a hat
```

contains two strings.

Wrong solution:

```
scanf("%s", s);
```

Problems with the above `scanf`: it won't read, and stops at, space characters like '` `', '`\t`' …

Strings input: method 2 = getchar()/mygetchar() (more complicated, but generous)

Given the declaration:

```
char s[MAXCHARS+1];
```

How to get `s` from standard input?

```
int c, i;
for (i=0; i<MAXCHARS
          && (c=getchar())!='\n'
          && c!=EOF                ; i++) {
   s[i]= c;
}
s[i]= '\0';
```

Strings: input − method 3 = …

there are other methods…

Some functions in <string.h>

```
strlen
strcpy
strcat
strchr
strstr
```

Strings: lec06 Exercise 3 + Exercise 5

Write a function `is_anagram(char *s1, char *s2)` that returns `1` if the two strings contain the same letters, possibly in a different order, and `0` otherwise, ignoring whitespace characters, and ignoring case. For example, `is_anagram("Algorithms", "Glamor Hits")` should return `1`.

How? if `s1` and `s2` have length `n` and `m`, what is the `big-O` complexity?

lec06. Exercise 1

Write a function `is_subsequence(char *s1, char *s2)` that returns `1` if the characters in `s1` appear within `s2` in the same order as they appear in `s1`. For example, `is_subsequence("bee", "abbreviate")` should be `1`, whereas `is_subsequence("bee", "acerbate")` should be `0`.

```
int is_subsequence(char *s1, char *s2) {

}
```

perhaps traverse s1, and with each of its chars `c`:

- ?

lec06. Exercise 1

Write a function `is_subsequence(char *s1, char *s2)` that returns `1` if the characters in `s1` appear within `s2` in the same order as they appear in `s1`. For example,
`is_subsequence("bee", "abbreviate")` should be `1`,
whereas `is_subsequence("bee", "acerbate")` should be `0`.

```
int is_subsequence(char *s1, char *s2) {
   char *p1, *p2= s2;
  for (p1= s1; *p1 ; p1++) {
     if (/* string p2 contains char *p1 */ ){
        ???

     } else ???
  }
  return ???
}
```

lec06. Exercise 1

Write a function `is_subsequence(char *s1, char *s2)` that returns `1` if the characters in `s1` appear within `s2` in the same order as they appear in `s1`. For example,
`is_subsequence("bee", "abbreviate")` should be `1`,
whereas `is_subsequence("bee", "acerbate")` should be `0`.

```
int is_subsequence(char *s1, char *s2) {
   char *p1, *p2= s2;
   for (p1= s1; *p1 ; p1++) {
      if ( (p2= strchr(p2, *p1) ) != NULL) {
         p2++;

      } else return 0;
   }
   return 1;
}
```

lec06: Exercise 1 and Exercise 5

```
int is_subsequence(char *s1, char *s2) {
   char *p1, *p2= s2;
  for (p1= s1; *p1 ; p1++) {
     if ( (p2= strchr(p2, *p1) ) != NULL) {
        p2++;
     } else return 0;
  }
  return 1;
}
```

If `n` and `m` are lengths of `s1` and `s2`, what is asymptotic performance of the above algorithm?

Assignment 1: What needs to be done today?

1. make a directory (say, `ass1`) for the assignment
2. copy all data files into `ass1`
3. build `ass1`.c from `ass1-skel.c`, add stage 0
4. compile and test
5. copy `ass1` to uni's `H:` if needed
6. compile and test if needed
7. `submit`
8. `verify`

NOTE: 15 minutes before end_of_class, do steps 5, 7, 8 regardless of the success of other steps.