卌

2510 : Lecture 16|| February 17, 2017

Assignment 2 is on Shareout!
- send email to c2510 @ haskell.bcit.ca
  Subject: <ID><lname><fname>

Review Exercise 1

```
1.c) void replace_last(char s[], int oldc, int new c) {
        size_t i;
        size_t t;
        for (i=0; s[i] != `\0'; i++) {
          if (s[i] == oldc)
              j=i; }
          if (j != (size_t)-1)
              s[j] = newc;
     }
```

```
2.c) void lowercase_copy (char dest [], const char src []) {
        size_t i;
        for (i=0;src[i]!='\0'; i++) {
            dest [i] = tolower (src[i]);
        }
        dest [i] = '\0';   /* need to add null char @ end */
     }
```

go backwards

```
  d) void reverse_copy (char dest [], const char src []) {
        size_t i; size_t j;
        for (i=strlen(src), j=0; i>0; i--, j++) {
            dest [j] = src [i-1];
        }
        dest [j] = '\0';
     }
```

...Lecture 16||

```
2.e) void strim_copy (char dest [], const char src []){
        size_t i; size_t j;
        for (i = Ø; src [i] != '\Ø'; i++){
            if (!isspace (src [i]))
                break;
        }
        for (j = Ø; src [i] != '\Ø'; i++, j++){
            dest [j] = src [i];
        }
        dest [j] = '\Ø';
     }
```

CAN BE REPLACED WITH strcpy (dest, & src [i]);

```
2.f) int is_valid_id (const char s []){
        if (strlen (s) != 9)
            return 0;
        ← if (s [Ø] != 'a' && s [Ø] != 'A')
            return 0;
        ← for (i = 1; i < 9; i++){
            if (!isdigit (s [i]))
                return 0;
        return 1;
     }
```

not indented

... Lecture 16||

4. a)

```
char a[] = "hello";
char * p = "world";
p = &a[1];    /* @ */
printf("%s", p);
```

Valid : ello

b)

```
char a[] = "hello";
char * p = "world";
p[1] = *a;    /* @ */
printf("%s", p);
```

invalid: can't change string constant

c)

```
char a[] = "hello";
char * p = "world";
a = p;  /* @ */
printf("%s", p);
```

invalid: can't assign to address (array name)

▷ Review Ex 2

1. a)

```
int i;
  for (i = 2; i < 10; i++){
    putchar('*');
    if (i > 7)
      break;
  }
```

2 → 8 7

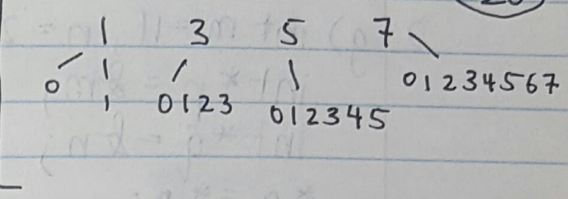
b)

```
int i;
  for (i = 1; i < 10; i++){
    if (i % 3 != 1)
      continue;
    putchar('*');
  }
```

$\overset{\downarrow}{1}$ 2 3 $\overset{\downarrow}{4}$ 5 6 $\overset{\downarrow}{7}$ 8 9 3x

...Lecture 16||

```
c) int i, j;
   for (i = 1; i < 8; i += 2){
     for (j = 0; j <= i; j++){
        putchar('*');
     }
   }
```



```
2.a) int n = 7;
     while (n > 2)
       printf("%d", --n);
```

6 5 4 3 2

```
b) int n = 7;
   while (n-- > 2)
     printf ("%d", n);
```

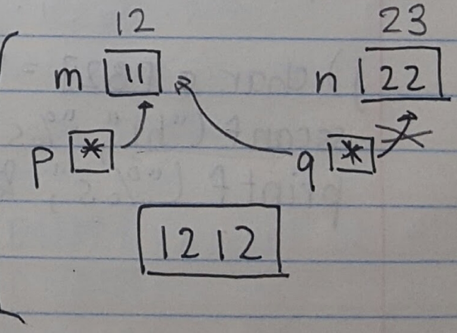
6 5 4 3 2

```
c) int n = 7
   while (--n > 2)
     printf ("%d", n);
```

6543

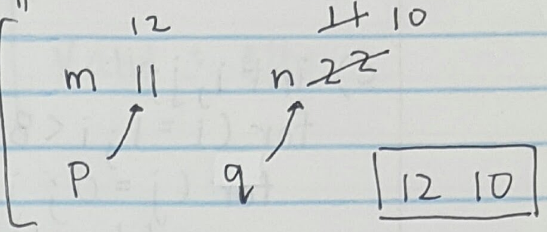
~~d) e)~~

```
f) int m = 11, n = 22;
   int *p = &m;
   int *q = &n;
   q = p;
   m++; n--;
   printf("%d %d", *p, *q);
```

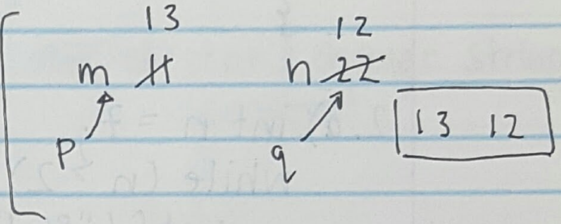


... Lecture 16

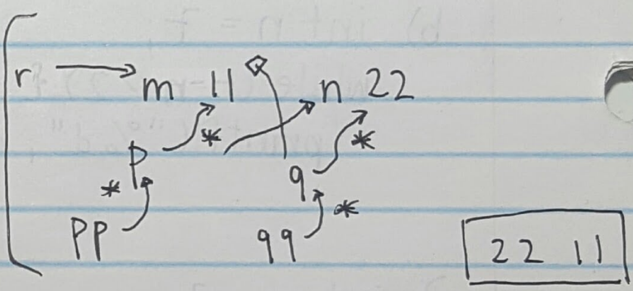
```
2. g) int m=11, n=22;
      int* p = &m;
      int* q = &n;
      *q = *p;
      m++; n--;
      printf("%d %d", *p, *q);
```



```
h) int m=11, n=22;
   int *p = &m;
   int *q = &n;
   *q = *p +1;
   *p = *q +1;
   printf("%d %d", m,n);
```



```
i) int m =11, n=22;
   int *p = &m, *q = &n;
   int **pp = &p, **qq = &q;
   int *r = ** pp;
   *pp = *qq;
   *qq = r;
   printf("%d %d", *p, *q);
```



```
j) char a[32] = "0123456789";
sscanf("hi", "%s", &a[4]);
printf("%s", &a[2]);
```

h i 1Ø

0 1 2 3 4 5 6 7 8 9

print: 2 3 h i

... Lecture 16||

3.a) int m = 4, n = 5, p = 6;

m = sscanf(" $\underbrace{12}$~~hello 34~~ ", "%d %d", &n, &p);

m	n	p
1	12	6

↑ unchanged

b) int m = 4, n = 5, p = 6;

m = sscanf("$\underbrace{12}\,\underbrace{-34}$.50", "%d %d", &n, &p);

m	n	p
2	12	-34

c) int m = 4, n = 5;

float f = 1.0;

m = sscanf("$\underbrace{34.50}$", "%f %d", &f, &n);

f	n	1
34.5	5	m

↑ unchanged

4. a) m = Øx89ab = 1000 1001 1010 1011

n = Øxef67 = 1110 1111 0110 0111

m&n = 1000 1001 0010 0011

= 8923

b) m = Øx89ab = 1000 1001 1010 1011

n = Øxef67 = 1110 1111 0110 0111

~m = 0111 0110 0101 0100

complement + or

|n 1111 1111 0111 0111

=FF77

c) m = Øx89ab = 1000 1001 1010 1011

n = Øxef67 = 1110 1111 0110 0111

XOR 0110 0110 1100 1100

(1 if bits are diff)
0 otherwise

= 66cc