WEEKLY-EXERCISE - 06

ICS 365-51 **Metropolitan State University/MN**

Week 7 **Due 11:59pm, Sunday, Oct. 9, 2022** **Fall 2022**

Name:_____Pong Lee_____ Score: _____

Please complete both Parts I and II and then upload the results to D2L under the dropbox for Weekly Exercise 05 before the deadline (total 20 points).

Part I: Based on the discussion in Lecture 6, please either **bold** or highlight your answers below, only one answer per question. (1 point each, total 10 points)

1. Based on the discussion on Python's List Comprehension in Chapter 6, which of the following lists is constructed by `[2 * x for x in range(5) if x % 2 == 0]` ?

A) `[0, 2, 4]`;
B) `[0, 2, 4, 8]`;
C) `[0, 4, 8]`;
D) `[0, 4, 8, 10]`;
E) Your answer: _____

2. Based on the discussion on Chapter 6 or slide 50 of Chapter 6, which of the following values is returned by `Lisp` statement `(list '1 (cdr '(1 2 3)))`?

A) `(1 2 3)`
B) `((1 2) 3)`
C) `(1 (2 3))`
D) `(1) (2 3)`
E) Your answer: _____

3. Based on the discussion in Chapter 6, which of the following statements is not true??

A) In Python, elements in a list can be of any data type, such as `myList = ["abc", 12, 34.5];`
B) Python's strings are mutable;
C) Python's lists are mutable;
D) A list in Python is similar to an array in Java.

4. Based on the discussion in Chapter 6, when a method with an object argument is invoked in Java, which of the followings is passed to the method?

A) The contents of the object
B) A copy of the object
C) The reference of the method
D) The reference of the object

5. 2. Based on the discussion on Chapter 6 or slide 50 of Chapter 6, which of the following values is returned by `Lisp` statement `(list '1 (car '(1 2 3)))`?

A) `(1 1)`
B) `(1 (1))`
C) `((1) 1)`
D) `(1 2)`
E) Your answer: _____

6. Based on the discussion on Chapter 7, which of the followings is the output of the C program on the left?

```
#include <stdio.h>

int main ()
{
int a = 2, b = 1;
if ((a < b) && (++b / 2))
  printf("A = %d\n", a);
else
  printf("B = %d\n", b);
}
```

A) A = 2
B) B = 1
C) B = 2
D) B = 3
E) Your answer: _____

7. Based on the discussion in Chapter 7, which of the followings is not one of the design issues for arithmetic expressions?

A) The operator-precedence rules;
B) Issues in operator overloading;
C) The number of operators allowed in a single arithmetic expression;
D) Whether type mixing is permitted in expressions.

8. Based on the discussion on Chapter 7, which of the followings is the output of the C program on the left?

```
#include <stdio.h>

int fun(int *i) {
  *i += 5;
  return 4;
}
void main () {
  int x = 1;
  x = fun(&x);
  printf("x = %d\n", x);
}
```

A) x = 4
B) x = 5
C) x = 6
D) x = 10
E) Your answer: _____

9. Based on the discussion on Chapter 7 or slide 9 of Chapter 7, which of the following values is returned by *Lisp* statement `(* 2 (* 2 (* 2 (+ 1 1))))?`

A) `2`
B) `4`
C) `8`
D) `16`
E) Your answer: _____

10. Based on the discussion in Chapter 7, which of the following statements is not true?

A) Precedence and associativity rules can be overridden with parentheses;
B) Use of an operator for more than one purpose is called operator overloading;
C) A narrowing conversion converts a value to a type that can include at least approximations of all of the values of the original type;
D) Arithmetic evaluation was one of the motivations for the development of the first programming languages.

Part II: Please study the discussion in class as well as covered in Chapters 6 and 7 of the textbook to complete the following tasks: (Total 10 points)

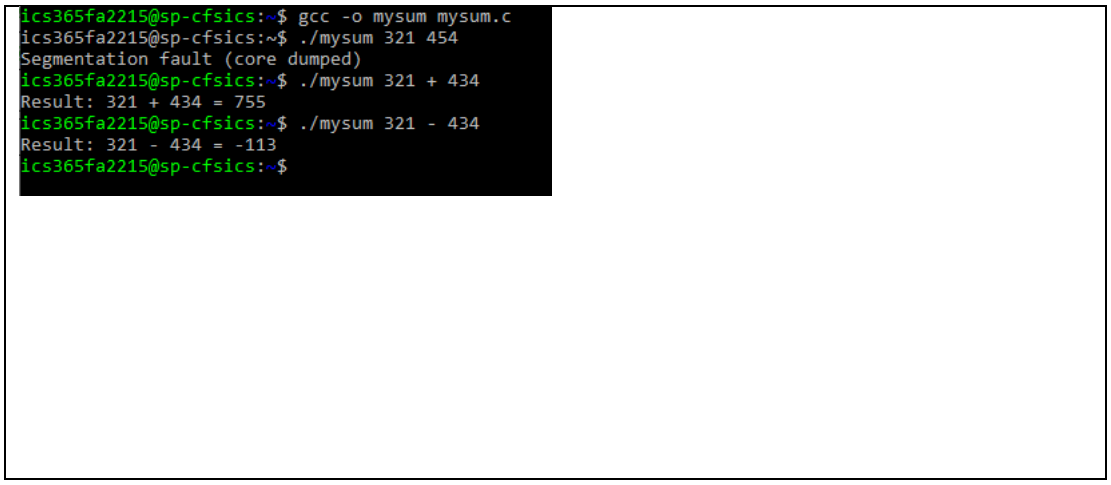
1. Given a **Perl** program below, please write a similar program in **C** on our Linux server, `sp-cfcsc01.metrostate.edu`. Please "`cat`" your program before executing it with the testing cases, and then include the corresponding screenshots below: (5 points)

A **Perl** program with its execution on two testing cases:

```
ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ cat mysum.pl
#!/usr/bin/perl -w
if ( $ARGV[1] eq "+" ) {
   $sum = $ARGV[0] + $ARGV[2];
   print "$ARGV[0] + $ARGV[2] = $sum \n";
} else {
   $sub = $ARGV[0] - $ARGV[2];
   print "$ARGV[0] - $ARGV[2] = $sub \n";
}
ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ ./mysum.pl 321 + 654
321 + 654 = 975
ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ ./mysum.pl 321 - 654
321 - 654 = -333
ics365fa2235@sp-cfsics:~/wk06$
```

Please provide the screenshot of a similar program in **C** with its execution on two testing cases:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char *argv[]){
        int a,b,result;
        char opt;
        a = atoi(argv[1]);
        b = atoi(argv[3]);
        opt = argv[2][0];
        switch(opt)
        {
                case '+':
                        result = a + b;
                        break;
                case '-':
                        result = a - b;
                        break;
                default:
                        result = 0;
                        break;
        }
        if(opt == '+' || opt == '-')
                printf("Result: %d %c %d = %d\n", a ,opt,b,result);

        return 0;
}
```

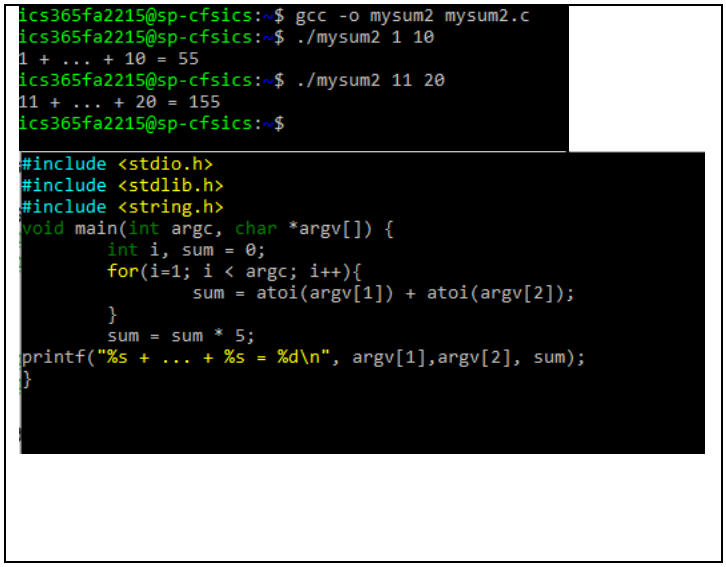

ics365fa2215@sp-cfsics:~$ gcc -o mysum mysum.c
ics365fa2215@sp-cfsics:~$ ./mysum 321 454
Segmentation fault (core dumped)
ics365fa2215@sp-cfsics:~$ ./mysum 321 + 434
Result: 321 + 434 = 755
ics365fa2215@sp-cfsics:~$ ./mysum 321 - 434
Result: 321 - 434 = -113
ics365fa2215@sp-cfsics:~$

2. Given a **Perl** program below, please write a similar program in **C** on our Linux server, `sp-cfcsc01.metrostate.edu`. Please "`cat`" your program before executing it with the testing cases, and then include the corresponding screenshots below: (5 points)

A **Perl** program with its execution on two testing cases:

```
ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ cat mysum2.pl
#!/usr/bin/perl -w
$sum = 0;
for ( $i = $ARGV[0]; $i <= $ARGV[1]; $i++ ) {
    $sum = $sum + $i;
}
print "\n$ARGV[0] + ... + $ARGV[1] = $sum \n\n"
ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ ./mysum2.pl 1 10

1 + ... + 10 = 55

ics365fa2235@sp-cfsics:~/wk06$
ics365fa2235@sp-cfsics:~/wk06$ ./mysum2.pl 11 20

11 + ... + 20 = 155

ics365fa2235@sp-cfsics:~/wk06$
```

Please provide the screenshot of a similar program in **C** with its execution on two testing cases:



```
ics365fa2215@sp-cfsics:~$ gcc -o mysum2 mysum2.c
ics365fa2215@sp-cfsics:~$ ./mysum2 1 10
1 + ... + 10 = 55
ics365fa2215@sp-cfsics:~$ ./mysum2 11 20
11 + ... + 20 = 155
ics365fa2215@sp-cfsics:~$
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void main(int argc, char *argv[]) {
        int i, sum = 0;
        for(i=1; i < argc; i++){
                sum = atoi(argv[1]) + atoi(argv[2]);
        }
        sum = sum * 5;
printf("%s + ... + %s = %d\n", argv[1],argv[2], sum);
}
```