

**Question 1: Identify syntax errors in the following program and fix them:**

```

1  #include <stdio.h>
2
3  void * addI (void * a, void * b);
4
5  int main () {
6
7      int * ans;
8      int c = 4, d = 10;
9
10     int choice;
11
12     void * (*f) (void *, void *) = addI;
13
14     ans = (int *) f(&c, &d);
15
16     printf ("%d \n", *ans);
17
18     return 0;
19 }
20
21 void * addI (void * a, void * b){
22
23     int * temp = malloc (sizeof (int));
24
25     *temp = (*a + *b);    *temp = *(int *)a + *(int *)b;
26
27     return temp;
28 }

```

Rules for Generic Pointers:

- \*a and \*b cannot be dereferenced
- needs to be casted and binded

**Question 2**

**2a:** Write a function prototype for mysteryWS10 and its definition using the code given below:

- mysteryWS10 takes 3 parameters: (a) int num1 (b) int num2 and (c) pointer to a function **fun** that takes 2 int parameters and returns an integer.
- mysteryWS10 returns the result of calling **fun** with parameters num1 and num2.

```

1  #include <stdio.h>
2
3  int sumOfSquares (int num1, int num2);
4  int sum (int num1, int num2);
5
6  /*
7  function prototype for mysteryWS10    int mysteryWS10(int a, int b, int(*f)(int, int));
8  */
9
10
11 int main () {
12
13     printf ("%d \n", mysteryWS10 (2, 10, sum));
14
15     printf ("%d \n", mysteryWS10 (2, 10, sumOfSquares));
16
17     printf ("%d \n", mysteryWS10 (2, sum (3, 4), sumOfSquares));
18
19     return 0;
20 }
21
22 int sum (int num1, int num2) {
23
24     return num1 + num2;
25 }
26
27 int sumOfSquares(int num1, int num2) {
28
29     return (num1 * num1 + num2 * num2);
30 }
31
32 /*
33 function definition for mysteryWS10    int mysteryWS10(int a, int b, int(*f)(int, int)) {
34                                         return (*f) (a,b);
35                                         // or
                                         return f(a,b);
                                         }

```

**2b.** What gets printed by lines 13, 15 and 17.

```

Line 13 - sum(a,b): 12
Line 15 - sumOfSquares(a,b): 104
Line 17 - 53

```

### Question 3:

3a. Complete the code given below by adding the prototype and function definition of function f2.

3b. Trace the code and show what gets printed

```

1 //Q3a Complete the code given below by adding the prototype and
  function definition of function f2
2 //Q3b trace the following code manually and show what gets printed
3 #include <stdio.h>
4
5 int f1 (int (*f) (int));
6 // add prototype of f2 here
7
8
9 int main () {
10
11     printf ("Answer = %d \n", f1 (f2));
12     return 0;
13 }
14
15 int f1 (int (*f) (int)) {
16
17     int n = 0;
18     while ((*f) (n)) {
19         n++;
20     }
21     return n;
22 }
23
24 //add function definition of f2 here
25
26
27

```

Function f2 takes an integer (say n), applies the following formula to it and returns the resulting value:  
 $n * n + n - 12$

$n = 0 \rightarrow \text{iteration \#1: } -12$   
 $n = 1 \rightarrow \text{iteration \#1: } -10$   
 $n = 2 \rightarrow \text{iteration \#1: } -6$   
 $n = 3 \rightarrow \text{iteration \#1: } 0$   
 $n = 3$

```

int f2(int n) {
    return n * n - 12;
}

```

**Question 4 (Optional):** Complete the function definition of dum. The call dum (f, i, j) should return f(i)+....+f(j). For example, (f, 1, 4) returns 30.

```

5 #include <stdio.h>
6
7 int dum (int (*f)(int), int start, int end);
8 int f (int i);
9
10 int main () {
11
12     printf ("Answer = %d \n", dum (f, 1, 4));
13     return 0;
14 }
15
16 int dum (int (*f)(int), int start, int end){
17
18     //add statements here
19 }
20
21 int f (int i) {
22
23     return i * i;
24 }

```

$f(1) + f(2) + f(3) + f(4)$   
 $1 + 4 + 9 + 16 = 30$

```

int sum = 0;
for (int i = start; i <= end; i++) {
    sum = sum + (*f) (i);    //or f(i)
}
return sum;

```