1. Determine the output for the program segment given in Table 1. Write your output with decimal points if applicable in **Output** column in Table 1. **(7 marks)**

Table 1

Line	Program Segment	Output
1	`int a, b = 2, c, d = 1, e;`	
2	`float p = 3.0, q;`	
3	`a = b * d++;`	
4	`c = ++a / 2 + d;`	
5	`b += a + c % 2;`	
6	`d *= (b - 1) / c;`	
7	`p = c * p / 5;`	
8	`e = --p + 2;`	
9	`q = static_cast<float>(e / 5) * d;`	
10		
11	`cout << "a = " << a << endl;`	a = 3
12	`cout << "b = " << b << endl;`	b = 6
13	`cout << "c = " << c << endl;`	c = 3
14	`cout << "d = " << d << endl;`	d = 2
15	`cout << "e = " << e << endl;`	e = 2
16	`cout << "p = " << p << endl;`	p = 0.8
17	`cout << "q = " << q << endl;`	q = 0

2. Given: `a = 3`, `b = 5`, and `c = 4`, determine the value of variable `z` for the following expressions. Label the order and results of execution for each operator in the boxes as stated in the expressions. The operator that is evaluated first should be labeled as 1, the second operator to be evaluated should be labeled as 2, and so on. Assume all variables of type `int`. For example: **(6 marks)**

3 6 4

	z	=	a	+	b	*	c	--	%	++	b
Order of execution				3		1		4	3	2	
Results of execution				5		20		c=3	2	b=6	

`z` = 5

z = 5

3. Determine the output for each code segment below:

(6 marks)

i.

```
int  n1 = 5, n2 = 10;
char code = 'A',code2 = 'Z';
bool t = true;


cout << (n1 + 3 == n2 + 5) << endl;
cout << ((n1 <= 5) && (n2 >= 10)) << endl;
cout << ((code == 'T') || (code2 != 'B')) << endl;
cout << ((t!=0) + 2 < 10) << endl;
```

0
1
1
1

ii.

```
bool b;
int x;
int y = 7;
b = y;
x = b;

cout <<"The value of x is " << x <<endl;
cout <<"The value of b is " << b <<endl;
```

The value of x is 1
The value of b is 1

4. Determine the output of each code segment below for the given value of `val = 4`.

(4 marks)

i.

```
int found = 0, count = val;

if (--count || !found == 0)
cout<< "danger" <<endl;
cout<< "count = " << count <<endl;
```

danger
count = 3

ii.

```
switch (val)
{
    case 10:
      cout<< "Perfect ";
      break;
    case 8:
      cout<< "Satisfactory ";
      break;
    default:
      cout<< "Unsatisfactory";
}
cout<< " : Pair Programming Evaluation";
```

Unsatisfactory : Pair Programming Evaluation

5. Complete program below to test the value of **a** and the program should display as below. **(5 marks)**

```
The given value is positive    or
The given value is negative    or
The given value is zero
```

```
int a; int b;

cout << "Please enter the value to be tested: ";
cin >> a;

 (a >= 0) ? (b = 1) : (b = 2);

 switch (b)
   { case 1: cout << " The given value is positive";
     break;
     case2: cout << " The given value is negative";
     break;
      default: cout<< The given value is zero";
      break;

   }
```

6. Based on the code given in Table 2, **(7 marks)**

Table 2

Line	Program Segment
1	`#include <iostream>`
2	`using namespace std;`
3	`int main()`
4	`{  for (int i = 5; i > 0; i =- 2)`
5	`  {  for (int j = 0; j =< i; j++)`
6	`  {  if (j%2)`
7	`  continue;`
8	`  else`
9	`  if !(j)`
10	`  cout << "i = " << i`
11	`  << ", j = << " j << endl;`
12	`  else`
13	`  break;`
14	`  }`
15	`  }`
16	`  return 0;`
17	`}`

a) Identify and fix the 4 errors in the code by writing the corrected statement in Table 3.

Answers:

Table 3

Line	Corrected Statement
4.	{for (int i=5; i>0 ; i-=2)
5.	{ for (int j=0, j<=i ;j++)
9.	if (!(j))
12.	<<" ,j" <<endl;

b) Assuming the errors are fixed, what is the output of the code?

Answers:

i=5, j=0
i=3, j=0
i=1, j=0

7. Given a flowchart in **Figure 1**. Complete and write the code segment of the flowchart using:

   i. `for` loop statement
   ii. post-test loop statement

   Note: Use a separate code segment for each question. **(15 marks)**

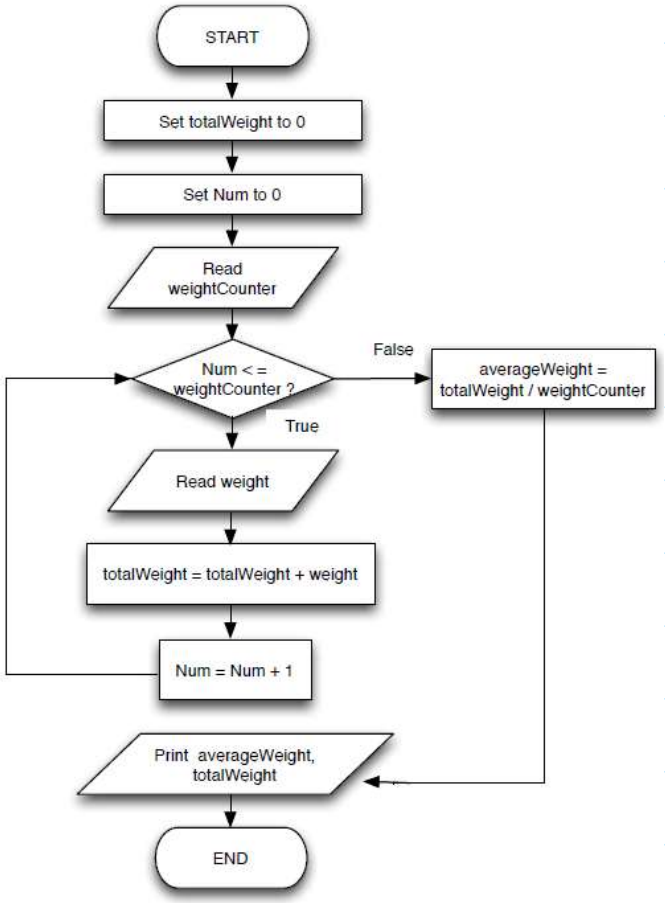
START
Set totalWeight to 0
Set Num to 0
Read weightCounter
Num < = weightCounter ?
False
averageWeight = totalWeight / weightCounter
True
Read weight
totalWeight = totalWeight + weight
Num = Num + 1
Print averageWeight, totalWeight
END

Figure 1

Answers:

i.

```
int weightCounter, totalWeight = 0;
cout << "Enter the number of students:  ";
cin >> weightCounter;
for (int num =0 , num <= WeightCounter , num++) {
  cout << " Enter  the weight : "
    float weight
     cin >> weight
    totalweight = totalweight + weight ;
    }
    float avgweight ;
    avgweight  = totalweight/weightCounter ;
    cout <<" Average  weight is " << avgweight << endl;
cout  << "total weight is " << total weight ;

  return 0;
```

ii.

```
int weightCounter, totalWeight = 0;
cout << "Enter the number of students:  ";
cin >> weightCounter;
do {
cout << " Weight = " ;
float weight, avgweight;
cin >> weight ;
totalWeight = totalWeight + weight;
 int num=0;
 num ++;
 } while (num <=weight Counter) ;
  avgweight = totalWeight / weight Counter
cout << " Average Weight is " << avgweight << endl;
cout << " Total weight is " << total weight;

 return 0;
```

8. **Program 1** below is meant to ask the user to enter two sides of a right triangle, **a** and **b** respectively, and display the length of hypotenuse. The length of hypotenuse is expressed by the following formula:

$$\sqrt{(a)^2 + (b)^2}$$

Complete **Program 1**, based on the instructions or comments written in (a) to (e).

(8 marks)

Line	Program 1
1	`#include <iostream>`
2	`#include <cmath>`
3	`using namespace std;`
4	`int main()`

```
4   int main()
5   {
6   // (a) Declare length for side a, side b and hypotenuse
7   float a,b, hypo                                    ;
8   // Display a corresponding prompt to the user
9    cout << "Enter the length of side a";
10  // (b) Input length side a
11    cin >> a                                         ;
12  // Display a corresponding prompt to the user
13    cout << "Enter the length of side b";
14  // (c) Input length side b
15    cin >>b                                          ;
16  // (d) Calculate the length of hypotenuse.
17    hypo = hypo (a,b)                                ;
18  // (e) Display the length of hypotenuse.
19    cout << hypo                                     ;
20    return 0;
21    }
```

9. Based on the output generated in **Output** column in Table 4, complete the blank spaces with appropriate predefined functions. Table 5 shows a list of predefined functions as a guide. **(7 Marks)**

Table 4

Line	C++ Statements	Output
1	`#include <iostream>`	
2	`#include <iomanip>`	
3	`#include <cstring>`	

2	`#include <iomanip>`	
3	`#include <cstring>`	
4	`#include <cctype>`	
5	`#include <cmath>`	
6	`using namespace std;`	
7	`int main() {`	
8	`  float num1 = 3.0, num2 = -2.5;`	
9	`  char word[15] = "#Programming!!", alp;`	
10	`  for (int i = 0; i <` 14 `; i++) {`	
11	`    alp =` toupper (word[i]) `;`	
12	`    cout << alp; }`	#PROGRAMMING!!
13	`  cout << endl;`	
14	`  cout <<` strcat (word, "**") `<< endl;`	#Programming!!**
15	`  cout << fixed << setprecision(3);`	
16	`  cout <<` pow (num2, num1) `<< endl;`	-15.625
17	`  cout <<` ceil (num2) `<< endl;`	-2.000
18	`  cout <<` abs (num2) `<< endl;`	2.500
19	`  return 0; }`	

Note : Manipulator `fixed` and `setprecision(3)` are to set the output of decimal precision fix to 3 decimal point.

Table 5

Predefined functions			
fabs (x)	abs (x)	ceil (x)	floor (x)
log (x)	pow (x,y)	sqrt (x)	exp (x)
strlen (x)	strcmp (x,y)	strstr (x,y)	pow10 (x)
strcat (x,y)	strcpy (x,y)	toupper (x)	tolower (x)
isupper (x)	islower (x)	isalpha (x)	isalnum (x)

10. Determine the output for the program segment given in Table 6. Write your output in **Output** column in the table. **(10 Marks)**

Table 6

Line	Program Segment	Output

Table 6

Line	Program Segment	Output
1	`void T1 (int &A, int B = 0)`	
2	`{`	
3	`  int C;`	
4	`  B = A-- + B - 2;`	
5	`  A += B;`	
6	`  C = A - B;`	
7	`  cout << A << " " << B << " " << C << endl;`	
8	`}`	
9		
10	`void T1 (int &A, int &B, int C, int D = 2)`	
11	`{`	
12	`  C += ++A - B;`	
13	`  B = 2 * C + D;`	
14	`  cout << A << " " << B << " " << C`	
15	`       << " " << D << endl;`	
16	`}`	
17		
18	`int main()`	
19	`{`	
20	`    int n1 = 3, n2 = 2;`	
21	`    T1(n2);`	1 0 1
22	`    T1(n1, n2);`	4 2 2
23	`    cout << n1 << " " << n2 << endl;`	4 1
24	`    T1(n2, n1, 2);`	2 2 0 2
25	`    cout << n1 << " " << n2 << endl;`	2 2
26	`    T1(n1, n2, n1-n2, 3);`	3 5 1 3
27	`    cout << n1 << " " << n2 << endl;`	3 5
28	`    return 0;`	
29	`}`	