Suppose **a, b** and **c** are integer variables that have been assigned the values **a = 9, b = 34** and **c =– 8.** Determine the value of each of the following arithmetic expressions.

1) b* (c % b)

2) a/c

3) a % c

4) a / b * c

A C program contains the following declarations and initial assignments:

```
int i= 1, j = 4, k;
float x = 0.007, y = -0.03, z;
char c = ' e ' , d = 'b ';
```

Determine the value of each of the following expressions. Use the values initially assigned to the variables for each expression.

5) (3 * j - 2 * i) % (2 * d + c)

6) 3 * ((i / 7) + (4 * (j - 2)) % (i + j - 4))

7) (x <= y) && (i < 0) || (j < 5)

8) z += (x >= 0) ? x : 0

11) i **/**= (j > 0) ? j : 0

9) (x == y) && (i > 0) && (j != 5)

10) k = (j ==5) ? i : j

A C program contains the following variable declarations.

float a = 9.5, b = 0.0003, c = 6100.;

Show the output resulting from each of the following **printf** statements.

Use underscores (_) to represent every intentionally left space.

12) printf("%f %f %f", a, b, c);

13) printf('%3f %3f %3f", a, b, c);

14) printf ("%8f %8f %8f", a, b, c);

15) printf("%l2.4e %12.4e %12.4e", a, b, c);

16) printf("%g %g %g", a, b, c);

Use the following to solve questions **17 – 19**

A C program contains the following statements:

```
#include <stdio. h>
int i, j, k;
```

Write **an** appropriate **scanf** function to enter numerical values for **i**, **j** and **k**, assuming the following

17) The values for **i**, **j** and **k** will be decimal integers not exceeding nine characters each.

18) The value for **i** will be a decimal integer, **j** an octal integer and **k** a hexadecimal integer, with each quantity not exceeding 8 characters.

19) The values for **i** and **j** will be octal integers and **k** will be a hexadecimal integer. Each quantity will be 6 or fewer characters

Use the following to solve questions **20 – 22**

A C program contains the following statements:

```
#include <stdio.h>
char text [ 60];
```

Write a **printf** function that will allow the contents of **text** to be displayed in the following ways.

20) Entirely on one line.

21) Only the first four characters.

22) The first four characters, preceded by eight blanks.

What output will be generated by the following C programs?

23)
```
#include <stdio.h>

main( )
{
  int i = 0, x = 0;

  while (i<10){
       if(i % 2 ==0) {

            x +=( i+1) ;

            printf("%d\t",
     x);
     }
       ++i;
   }
  printf("\nx = %d ", x);

    }
```

24)
```
#include <stdio.h>

main( )
{
  int i = 0, x = 1;

  while (i<10){
       if(i % 2 ==1) {

            x += i;

            printf("%d, %d
     \t", x, i);
     }
       ++i;
   }
  printf("\nx = %d", x);

    }
```

25)
```
#include <stdio.h>

main( )
{
  int i = 1, x = 0;

  while (i<10){
       if(i % 5 ==0) {

            x += i;

            printf("%d, %d
     \t", x, i);
     }
       ++i;
   }
  printf("\nx = %d", x);

    }
```

26)
```
#include <stdio.h>
            main( )
       {
     int i = 0, x = 0;
     for (i = 1; i < 5; ++i)
     {
      if (i % 2 == 1)
          x += i;
        else
            x--;
        printf("%d, %d \t",x,i);
     }
         printf("\nx = %d\t ", x
      );
       }
```

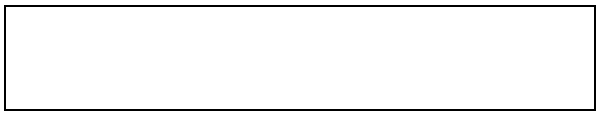
27)
```
#include <stdio.h>
            main( )
       {
     int i = 0, x = 0;
     for (i = 1; i < 5; ++i)
     {
      if (i % 2 == 1)
          x += i;
        else
            x--;
        printf("%d, %d \t", x,
i);
        continue;
     }
         printf("\nx = %d\t ", x
      );
       }
```

28)
```
#include <stdio.h>
              main( )
         {
     int i = 0, x = 0;
     for (i = 1; i < 5; ++i)
     {
      if (i % 2 == 1)
          x += i;
        else
              x--;
        printf("%d, %d \t", x,
i);
        break;
     }
         printf("\nx = %d ", x
      );
        }
```

29)
```
#include <stdio.h>
   main( )
   {
     int i = 0, x = 0;
     for (i = 1; i < 10; i *= 2)
     {
         x++;
         printf("%d,  %d  \t",  x,
     i);
     }
     printf("\nx = %d", x);
       }
```

30)
```
#include <stdio. h>
       int funct1 (int count);
       main( )
       {
          int a, count;
          for (count = 1; count
   <= 5; ++count) {
        a = funct1(count);
        printf ( "%d\t ", a ) ;
             }
         }

   int funct1(int x)
         {
        int y = 0;
        y = -1 * ( x * x) % -3 ;
        return(y);
           }
```



31)
```
#include <stdio.h>
         int funct1 ( int
count);
    main( )
    {
        int a, count;
        for (count = 1; count <=
   3; ++count)
        {
              a = funct1(count);
       printf("%d, %d\t ", a,
   count);
        }
     }

     funct1 (int x)
     {
          int y = 0;
                y += x;
           return(y);
     }
```

32)
```
#include <stdio. h>
int a = 10, b = 20;
int funct1(int a, int b);
main( )
{
  int count, c, d;
  for (count = 1; count < 3;
++count){
      c = 20 * (count - 1);
      d = 4 * count * count;
      printf("%d %d\t " ,
funct1(a,c) , funct1(b,d));
      }
  }

funct1(int x, int y)
{
   return(x - y);
}
```

33)
```
#include <stdio.h>
main ( )
{
  int a, b = 0;
  int c[10]={2, 2, 4, 4, 6 , 6,
7, 8 , 9, 0};
  for (a = 0; a < 10; ++a)
     if ((c[a] % 2) == 0)
        b += c[a];
  printf('%d', b);
 }
```

34)
```
#include <stdio.h>
main ( )
{
  int a, b = 0;
  int c[10]={1, 1, 3, 3, 5 , 5,
7, 8 , 9, 10};
  for (a = 0; a < 10; ++a)
     if ((c[a] % 2) == 1)
        b += c[a];
  printf('%d', b);
 }
```

35)
```
#include <stdio.h>
main ( )
{
   int a, b = 0;
   int c[5] = {1, 2, 3, 4, 5 };
   for (a = 0; a < 5; ++a)
         if ((a % 2) == 1)
            b += c[a];
   printf("%d", b);
 }
```

36)
```
#include <stdio.h>
#define ROWS  3
#define COLUMNS  4
int z[ROWS][COLUMNS] = {1, 2, 3, 4, 5 , 6, 7,
                  8, 9, 10, 11, 12};
main ( )
{
     int a, b, c = 999;
     for(a = ROWS-1; a >=0; --a)
       for(b =COLUMNS-1; b >= 0; --b)
        if( z [ a ] [ b ] < c)

                        c = z[a][b];
     printf ( " % d " , c);
}
```

37)
```
#include <stdio.h>
#define ROWS  3
#define COLUMNS  4
int z[ROWS][COLUMNS] = {1, 2, 3, 4, 5 , 6, 7,
                  8, 9, 10, 11, 12};
main ( )
{
     int a, b, c = 999;
     for(a = ROWS-1; a >=0; --a)
       for(b =COLUMNS-1; b >= 0; --b)
        if( z [ a ] [ b ] < c){
                        c = z[a][b];
           printf ( " % d " , c);
        }
}
```