KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

COLLEGE OF ENGINEERING

BSc. BIOMEDICAL/ELECTRICAL & TELECOMMUNICATION ENGINEERING

INTRODUCTION TO C−PROGRAMMING (COE 251)

FIRST SEMESTER EXAMINATION, 2017

DURATION: 2.5 HOURS

INDEX NUMBER:…………………………………….... PROGRAMME:……………………………………………

Suppose **a, b** and **c** are integer variables that have been assigned the values **a = 8, b = 3** and **c =−5**. Determine the value of each of the following arithmetic expressions.

1) 2 * b + 3 % (a + c)

2) a * b / c

3) a * (c % (−b))

4) 2 * b + 3 % (a / c)

A C program contains the following declarations and initial assignments:

int i= 9, j = 7, k = 1;
float x = 0.003, y = −0.03, z;
char c = ' A ' , d = 'B ';

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

5) (3 * i – 2 * j) % (2 * d – c)

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

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

8) (x < y) && (i > 0) && (j < 5)

9) k *= (j !=5) ? i : j

10) z = (x <= 0) ? x : 0

11) i –= (j < 0) ? j : 0

A C program contains the following variable declarations.

float a = 6.5, b = 0.0365, c = 6000.;

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

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

13)printf('%4f %4f %4f", a, b, c);

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

15)printf("%8.1e %8.1e %8.1e", a, b, c);

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

17) printf("%08g %08g %08g", a, b, c);

A C program contains the following statements:

#include <stdio.h>

char a, b, c;

Use the program above to answer questions **18 –19**.

18)Write appropriate **getchar** statements that will allow values for **a**, **b** and **c** to be entered into the computer.

19) Write appropriate **putchar** statements that will allow the current values of **a**, **b** and **c** to be written out of the computer (i.e., to be displayed)

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

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

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

21) 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.

22) The values for **i** and **j** will be hexadecimal integers and **k** will be an octal integer. Each quantity will be 5 or fewer characters

Use the following to solve questions **23 – 26**

A C program contains the following statements:

```
#include <stdio.h>
char text [ 40];
```

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

23) Entirely on one line.

24) Only the first eight characters.

25) The first eight characters, preceded by five blanks.

26) The first eight characters, followed by five blanks.

What output will be generated by the following C programs?

27)
```
#include <stdio.h>
   int main( )
   {
       int i = 0, x = 0;
       while (i<=10){
          if(i % 2 ==0) {
             x += i;
             printf("%d\t", x);
            }
            ++i;
        }
       printf("\nx = %d ", x);
       return 0;
   }
```

28)
```
#include <stdio.h>
   int main( )
   {
       int i = 0, x = 0;
       while (i<=10){
          if(i % 2 ==1) {
             x += i;
             printf("%d\t", x);
            }
            ++i;
        }
       printf("\nx = %d", x);
       return 0;
   }
```

29)
```
#include <stdio.h>
   int main( )
   {
       int i = 0, x = 0;
       while (i<10){
          if(i % 5 ==0) {
             x += i;
             printf("%d\t", x);
            }
            ++i;
        }
       printf("\nx = %d", x);
       return 0;
   }
```

30)
```
#include <stdio.h>
    int main( )
    {
       int i = 0, x = 0;
       for (i = 1; i < 7; ++i)
       {
         if (i % 2 == 1)
              x += i;
         else
              x−−;
         printf ("%d\t", x);
       }
       printf("\nx = %d\t ", x );
       return 0;
    }
```

```
31)#include <stdio.h>
     int main( )
    {
        int i = 0, x = 0;
        for (i = 1; i < 7; ++i)
        {
          if (i % 2 == 1)
                 x += i;
          else
                 x−−;
          printf ("%d\t ", x);
          continue;
        }
        printf(“\nx = %d\t ", x );
        return 0;
    }
```

```
32)#include <stdio.h>
     int main( )
    {
        int i = 0, x = 0;
        for (i = 1; i < 7; ++i)
        {
          if (i % 2 == 1)
                 x += i;
          else
                 x−−;
          printf ("%d\t ", x);
          break;
        }
        printf(“\nx = %d ", x );
        return 0;
    }
```

```
33)  #include <stdio.h>
      int main( )
      {
         int i = 0, x = 0;
         for (i = 1; i < =9; i *= 2) {
           x++;
           printf("%d\t ", x);
         }
         printf("\nx = %d", x);
         return 0;
      }
```

```
34)  #include <stdio.h>
      int main( )
      {
         int i = 0, x = 0;
         for (i = 1; i <=9; i *= 2) {
           x++;
           printf("%d\t ", i);
         }
         printf("\nx = %d", x);
         return 0;
      }
```

```
35)  #include <stdio. h>
      int funct1 (int count);
      int main( )
      {
         int a, count;
         for (count = 1; count <= 4; ++count) {
              a = funct1(count);
              printf ( "%d\t ", a ) ;
           }
          return 0;
        }

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

```
36)  #include <stdio.h>
      int funct1 ( int count);
      int main( )
      {
         int a, count;
         for (count = 1; count <= 4; ++count)
         {
             a = funct1(count);
              printf("%d\t", a);
         }
          return 0;
       }

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

```
37)#include <stdio.h>
      int funct1( int count);
      main ( )
      {
         int a, count;
         for(count = 1; count <= 4; ++count)
         {
             a = funct1(count);
             printf("% d\t", a);
         }
       return 0;
      }
     int funct1( int x)
     {
       static int y = 0;
       y += x;
       return(y);
      }
```

```
38)#include <stdio. h>
       int a = 15, b = 25;
       int funct1(int a, int b);
       int main( )
       {
         int count, c, d;
         for (count = 1; count < 4; ++count){
            c = 15 * (count − 1);
            d = 5 * count * count;
            printf("%d %d\t " , funct1(a,c) , funct1(b,d));
            }
          return 0;
         }

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

39)
```
 #include <stdio.h>
  int funct1(int a);
  int funct2(int a);
  int main( )
  {
   int a = 0, b = 1, count;
   for (count = 1; count <= 3; ++count) {
        b += funct1(a) + funct2(a);
        printf( " % d\t", b);
   }
  return 0;
  }

int funct1( int a)
 {
     int b;
     b = funct2(a);
     return(b);
 }

int funct2(int a)
{
    static int b = 1;
    b += 1;
    return(b + a);
}
```

40)
```
 #include <stdio.h>
   int main ( )
   {
    int a, b = 0;
    static int c[10]={1, 2, 3, 4, 5 , 6, 7, 8 , 9, 0};
    for (a = 0; a < 10; ++a)
      if ((c[a] % 2) == 0)
        b += c[a];
    printf('%d', b);
    return 0;
   }
```

41)
```
 #include <stdio.h>
   int main ( )
   {
    int a, b = 0;
    static int c[10]={1, 2, 3, 4, 5 , 6, 7, 8 , 9, 0};
    for (a = 0; a < 10; ++a)
      if ((c[a] % 2) == 1)
        b += c[a];
    printf(“%d”, b);
   return 0;
   }
```

42)
```
#include <stdio.h>
   int main ( )
   {
     int a, b = 0;
     static int c[10] = {1, 2, 3, 4, 5 , 6, 7, 8 , 9, 0};
     for (a = 0; a < 10; ++a)
       if ((a % 2) == 0)
          b += c[a];
     printf(“%d”, b);
     return 0;
   }
```

43)
```
 #include <stdio.h>
   int main ( )
   {
     int a, b = 0;
     static int c[10] = {1, 2, 3, 4, 5 , 6, 7, 8 , 9, 0};
     for (a = 0; a < 10; ++a)
       if ((a % 2) == 1)
         b += c[a];
     printf(“%d”, b);
     return 0;
   }
```

44)
```
#include <stdio.h>
  int main ( )
  {
    int a, b = 0;
    int c[10] = {1, 2, 3, 4, 5 , 6, 7, 8, 9, 0};
    for (a = 0; a < 10; ++a)
       b += c[a];
    printf( "%d", b) ;
  }
```

45)
```
#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};
 int main ( )
 {
         int a, b, c = 999;
         for(a = 0; a < ROWS; ++a)
           for(b = 0; b < COLUMNS; ++b)
               if( z [ a ] [ b ] < c)
                  c = z[a][b];
         printf ( " % d " , c);
 }
```

Use the following to solve questions **46 – 53**

A C program contains the following statements.

```
      int i, j = 25;
      int * pi , * pj = &j;
     *pj = j + 5;
      i   = *pj + 5;
      pi = pj ;
     *pi = i+ j;
```

Suppose each integer quantity occupies 2 bytes of memory. If the value assigned to **i** begins at (hexadecimal) address **F9D** and the value assigned to **j** begins at address **F9F**, then

46)What value is represented by **&i**?

47)What value is represented by **&j** ?

48)What value is assigned to **pj** ?

49)What value is assigned to ***pj** ?

50)What value is assigned to **i** ?

51)What value is represented by **pi**?

52)What final value is assigned to ***pi**?

53)What value is represented by the expression **(*p i + 2)?**

Use the following to solve questions **53 – 60**

A C program contains the following statements.

```
char u, v = 'A' ;
char *pu, *pv = &v;
……
*pv = v + 1;
  u = *pv + 1;
 pu = &u;
```

Suppose each character occupies 1 byte of memory. If the value assigned to **u** is stored in (hexadecimal) address **F8C** and the value assigned to v is stored in address **F8D**, then

54)What value is represented by **&v**?

55)What value is assigned to **pv**?

56)What value is represented by *pv?

57)What value is assigned to **u**?

58)What value is represented by **&u**?

59)What value is assigned to **pu**?

60)What value is represented by ***pu**?

Use the following to solve questions **61 – 69**

A C program contains the following statements.

```
float a = 0.001, b = 0.003;
float c, *pa, *pb;
pa  = &a;
*pa = 2 * a;
pb  = &b;
c   = 3 * (*pb – *pa);
```

Suppose each floating–point number occupies **4** bytes of memory. If the value assigned to **a** begins at (hexadecimal) address **1130**, the value assigned to **b** begins at address **1134**, and the value assigned to **c** begins at **1138**, then

61)What value is assigned to **&a**?

62)What value is assigned to **&b**?

63)What value is assigned to **&c**?

64)What value is assigned to **pa**?

65)What value is represented by ***pa**?

66)What value is represented by **&(*pa)**?

67)What value is assigned to **pb**?

68)What value is represented by ***pb**?

69)What value is assigned to **c**?

Use the following to solve questions **70 – 71**

A C program contains the following statements.

static int x[8] = {10, 20, 30, 40, 50, 60, 70, 80};

70)What is the value of **(*x + 2)**?

71)What is the value of ***(x + 2)?**

State whether the following are **true** or **false**

72) The amount of white (blank) space you leave in C does not affect how the code is executed?

a) True
b) False

73)In the **for** loop we are always guaranteed of at least one iteration.

a) True
b) False

74)A function definition is also known as a function prototype.

a) True
b) False

75)In the Arduino IDE the pre–loaded Examples are not editable?

a) True
b) False

INDEX NUMBER:……………………………………………………………………………………………

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK FOR YOUR ROUGH WORK. PLEASE DO NOT DETACH IT.

Stay Blessed!