Practice Questions for C Programming

Prepared by Dr. Sufyan Almajali

جامعــة الأمـــيرة ســـمية للتكـنولـوجـيا

Princess Sumaya University for Technology

Note: This study guide includes a list of practice questions and their answers, the purpose of this guide to help you practice via question and answer approach. The guide by itself is not enough, you need to study the book, the slides and lecture notes.

Q1) What is the output of the following program, assuming that the address of x is `003674D0`, `the address of a is 00367038`?

```
#include <stdio.h>

int *p, x,y;
int a[5]={ 100,200,300,400,500};
int *p2;
int main()
{
      p=NULL;
      x=10;
      p=&x;
      printf("1) %d %d %p %p %p \n",x,*p,p,&x,&p);
      p2=&x;
      printf("2) %d %d %p %p  \n",x,*p2,p2,&x);
      p2=a;
      printf("3) %d %d %p %p  \n",a[0],*p2,p2,a);
      p2=&a[2];
      printf("4) %d %p %p  \n",*p2,p2,a);
      p2++;
      printf("5) %d %p   \n",*p2,p2);
      p=a;
      y=*(p+2);
      printf("6) %d %d   \n",y,a[2]);
      printf(" loop using pointer \n");
      for ( int * p3=a; p3 <= &a[4]; p3++)
            printf("loop: %d %p \n",*p3,p3);
      x=9;
      int * const p4=&x;
      *p4=5;
      printf("7) %d %d   \n",x,*p4);

      // p4=&y; error to change the value of p4 as it is a constant pointer
      // array is a constant pointer too
```

```
return 0;
}
```

Q2) What will be the output of the following program?

```
#include <stdio.h>

int *p, x;
int a[5]={100,200,300,400,500};
int *p2;
int main()
{
      p=NULL;
      x=500;
      p=&x;
      printf("1) %d %d \n",x,*p,p,&x,&p);
      p2=a;
      *(p2+1)=*p;
      *p= *p2 + *(p2+2);
      printf("2) %d %d \n",x,*p,*p2);

      // p4=&y; error to change the value of p4 as it is a constant pointer
      // array is a constant pointer too
return 0;
}
```

Q3) What is the output of the following program

```
#include <stdio.h>
int z=0;

int f( int &x, int y, int * m)
{ int static kk=9;

x=x+2;
y=y+3;
z=z+y+1+*m;
kk=kk+x;
*m=kk;
printf("%d %d %d %d %d \n", x,y,kk,z,*m);
return x+y;
}

int main()
{
int k=3,m=5,r=0;
printf("%d %d %d %d  \n", k, m,r, z);
z=f(k,m,&r);
printf("%d %d %d %d  \n", k, m,r, z);
z=f(m,k,&r);
printf("%d %d %d %d  \n", k, m,r, z);

return 0;
}
```

Q4) what is the output of the following?

```
#include <stdio.h>
int f1( int x, int y)
{
x=x+2;
y=y+3;
return x+y;
}
int f2( int &x, int y)
{
x=x+2;
y=y+3;
return x+y;
}
int f3 ( int * x, int * y)
{
*x=*x+2;
*y=*y+3;
return *x+*y;
}
int f4( int x, int &y, int * z)
{
x=x+y;
y=*z+3;
z=&x;
*z=y*2;
return *z;
}

int main()
{
int k=3,m=5,r=0;
printf("1) %d %d %d \n",k,m,r);
r=f1(k,m);
printf("2) %d %d %d \n",k,m,r);
r=f2(k,m);
printf("3) %d %d %d \n",k,m,r);
r=f3(&k,&m);
printf("4) %d %d %d \n",k,m,r);
r=f4(k,m,&r);
printf("5) %d %d %d \n",k,m,r);

return 0;
}
```

Q5)

```
#include <stdio.h>

int main()
{
      int a[4][5]={ {1,2,3}, {2}, {1,2,2,2,2}, {32,23}};
int i,j;

for (i=0;i<4;i++)
      for (j=0;j<4;j++)
       a[i][j]=a[i][j+1]+i*j+i+j;


for (i=0;i<4;i++)
{
      for (j=0;j<5;j++)
      printf(" %d",a[i][j]);

      printf("\n");
}

return 0;
}
```

Q6)

```
#include <stdio.h>

void f1( int &x, int y)
{
      x=x+y;
}

void f2 ( int d[][5], int ROWS)
{

int i,j;

for (i=0;i<4;i++)
      for (j=0;j<4;j++)
       d[i][j]=d[i][j+1]+3;

}
int main()
{
      int a[4][5]={ {11,21,3}, {2,3,3}, {7,12,2,2,2}, {32,23}};
      int i,j;

f1(a[2][1],a[3][0]);

f2(a,4);

for (i=0;i<4;i++)
{
      for (j=0;j<5;j++)
      printf(" %d",a[i][j]);

      printf("\n");
}

return 0;
}
```

Q7)

```
#include <stdio.h>

void f1( int &x, int * y, int z)
{
      x=x+*y;
      *y=x+2;
      z=z+3;
}

void f2 ( int d[], int SIZE)
{
int i;
for (i=0;i<SIZE;i++)
       d[i]=d[i]*2;

}
int main()
{
      int a[4]={0};
      int b[5]= {1, 2};

      int i;

f1(b[2],&b[1],b[0]);
f1(a[0],&b[1],b[2]);

f2(a,4);
f2(b,4);

for (i=0;i<4;i++)
      printf(" %d",a[i]);
printf("\n");


for (i=0;i<5;i++)
      printf(" %d",b[i]);
printf("\n");

return 0;
}
```

Q8)
What is the output of the following program:

```
#include <stdio.h>
int z=0;

int f( int &x, int * m)
{ int static kk=10;
x=x+2;
z=z+*m;
kk=kk+x;
*m=kk;
return x+kk;
}

int main()
{
int k=3,m=5,r=0, *p;
z=f(k,&r);
printf("%d %d %d %d  \n", k, m,r, z);
p=&m;
z=f(m,&r);
printf("%d %d %d %d  %d  \n", k, m,r, z,*p);

return 0;
}
```

Q9)

```
#include <stdio.h>
int z;

int f1( int x, int y)
{
x=x+2;
y=y+3;
z=y+3;
return x+y;
}
int f2( int &x, int y)
{
x=x+2;
y=y+3;
return x+y;
}
int f3 ( int * x, int * y)
{
*x=*x+2;
*y=*y+3;
return *x+*y;
}
int f4( int x, int &y, int * z)
{
x=x+y;
y=*z+3;
z=&x;
*z=y*2;
return *z;
}

int main()
{
      z=10;
int k=3,m=5,r=0;
printf("1) %d %d %d %d \n",k,m,r,z);
r=f1(k,m);
printf("2) %d %d %d %d \n",k,m,r,z);
r=f2(k,m);
printf("3) %d %d %d %d \n",k,m,r,z);
r=f3(&k,&m);
printf("4) %d %d %d %d \n",k,m,r,z);
r=f4(k,m,&r);
printf("5) %d %d %d %d \n",k,m,r,z);

return 0;
}
```

Q10)

Assume the following was the input to below program, what will be the output?

```
10 20 30 1 2 3 4 5 6
5 5 5 6 6 6 7 7 7 8 8 8
9 9 9
10 10 10
```

```
#include<stdio.h>

void Print2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
      {
            for (int c=0; c<3;c++)
                  printf("%d ",a[r][c]);
            printf("\n");
      }
}

void Read2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                  scanf("%d ",&a[r][c]);
}

void Process2DArrays( int  a[][3], int  b[][3], int  m[][3], int ROWS)
{
      for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                        m[r][c]=a[r][c]*b[r][c]+r+c;

}
```

```
int main()
{
      int x[4][3],y[4][3],z[4][3];
      Read2DArray(x,4);
      Read2DArray(y,4);
      Process2DArrays(x,y,z,4);
      printf("================\n");
      Print2DArray(z,4);

      return 0;
}
```

Q11) Write a C Program that declares 3 arrays of size 4 by 3 and type integer. Your program should read the values of the first two arrays, add the two arrays values and store the results in a third array. Declare function to read the array values, use the function twice. Declare another function to add two arrays and store the result in a third one.

Q12) what is the output of the following:

```
#include <stdio.h>
int z;

int f1( int x)
{
   if (x<=4)
         return 8;
   else
         return x+f1(x-1);
}

int f2( int &x, int & y)
{
x=x+1;
y=y+2;
return x+y;
}

int main()
{

int k=3,m=5,r=0;
printf("1) %d %d  \n",f1(k),f2(m,r));
r=f1(m);
printf("2) %d %d \n",r,f2(r,r));
k=f1(r-20);
printf("3) %d %d \n",k,r);

return 0;
}
```

Q13)A. Write a C function that computes that maximum of a 2D array of size 6 by 5.

Q13) B. Write a C function that computes that total sum of a 2D array of size 6 by 5.

Q14) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return true if found false otherwise.

Q15) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return the row # of location the value was found at if found and return -1 if not found.

Q16) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return the row # and col# of location the value was found at if found and return -1 if not found. The function returns these two valeus using arguments passed by reference.

Q17) Write a C function that computes that maximum of a specific row R in a 2D array of size 6 by 5.

Q18) Write a C function that computes that total sum of of a specific col C in a 2D array of size 6 by 5.

Q19) Write a C function that searches for value key in a specific row R in a 2D array of size 6 by 5. The function should return true if found false otherwise.

Q20) What is the output of the following program

```
#include<stdio.h>


int main()
{
      char a[4][15]= { "PSUT","C Programming","Data","Structure"};
      for (int i=0;i <4;i++)
            printf(" %s \n", a[i]);


      a[2][2]=a[0][1];
      a[3][1]=a[1][2];
      a[0][3]=a[3][1];

            for (int i=0;i <4;i++)
            printf(" %s \n", a[i]);

      return 0;
}
```

Q21)

```
#include<stdio.h>


int main()
{
      char a[4][15]= { "PSUT","C Programming","Data","Structure"};
      char *p;
      for (int i=0;i <4;i++)
            printf(" %s \n", a[i]);


      p=&a[0][0];

      printf(" %c \n", *p);
      p=p+1;
      printf(" %c \n", *p);

      p=&a[2][0];
      printf(" %c \n", *p);
      p=p+1;
      printf(" %c \n", *p);



      return 0;
}
```

Q22)

Determine the value of the following expressions.

a. toupper('b')
b. tolower('C');
c. pow(3.0,3.0);
d. sqrt(81.0);
e. fabs(-1.23);
f. floor(22.46);
g. ceil(33.3);

Q23)

. Using the functions in the cmath library, write the following mathematical formulas as C expressions.

a. $3.0^{2.4}$
b. $(x - y)^{1/2}$ Note: the ½ power is the square root

c. |y - 42.3|

d.$(-b + (b^2 - 4ac)^{1/2}) / 2a$

Q24)

Write a C function that has an input of a char value and returns true if the character is lower case or false otherwise.

Q25)

Write a C function that has three inputs which are integers. The function returns true if the first number raised to the power of the second number equals the third number.

Q26)

What is the output of the following program fragment:

```
int find(int a, int b, int c)
{
        int temp;

        c = a + b;

        temp = a;

        a = b;

        b = 2 * temp;

        printf(“%d  %d  %d \n”,a,b,c);

        return b;



}

int main()

{

        int x, y, z;

        x = 15;

        y = 25;

        z = 30;

        printf(“%d  %d  %d %d \n”,          x,y,z,find(x,y,z));
}
```

Q27)

Write C statements to do the following:

a. Declare an array alpha of 17 components of type int.

b. Output the value of the 12th component of the alpha array.

c. Set the value of the 5th component of the alpha array to 35.

d. Set the value of the 9th component of the alpha array to the sum of the 6th and 13th components of the alpha array.

e. Set the value of the 4th component of the alpha array to three times the value of the 8th component minus 57.

f. Output alpha so that five components appear on each line.

Q28)

What is stored in the list array after the following code executes?

```
int list[6];
      for(int j = 0; j < 6; j++)
      {
            list[j] = 3 * j + 4;
            if (j % 3 == 0)
            {
                  list[j] = list[j] - 2;
            }
      }
```

Q29) Write C statements to define and initialize the following arrays using appropriate data types:

a. An array of grades of size 10 initialized all to 0

b. An array of weights has 4 components which have the following values: 5.05, 5.8, 6.3, 6.6.

c. An array of ages has 6 components which have the values: 80,60,21,42,22,21.

d. An array of symbols which contains the following characters: '$', '%', '@', '!', '|', '&'.

Q30) Given the following declaration, what is stored in the 8th element of the array?

```
int list[10] = {1, 2, 3, 4, 5};
```

Q31) Write a for loop to initialize the following array (int data[10]) with the values 10, 9, 8… 1.

Q32) Given an array of 10 doubles named data, write a loop that loads the array with user input.

Q33) Given an array of 100 doubles named data, write a loop that creates the sum of all the array elements.

Q34) Write a loop that finds the smallest element in an integer array called data containing 100 elements.

Q35) Write a code fragment using ONE for-loop to calculate the sum and product of the elements of a floating point array. The array is declared as shown below. Include any other declarations and initialization statements necessary to perform the operation. Display the sum and product results after the for-loop using *printf*.

```
const int MAX = 15;
float values[MAX] = {98.0, 75.0, 66.5, 78.5, 83.5, 89.0,
   94.0, 43.0, 99.0, 88.0, 42.5, 72.0, 59.5, 77.0, 63.5};
```

Q36) Write a function to have a user enter some number of integers into an array. The integer values must be between -100 and +100 inclusive (+100 and -100 should be accepted as valid inputs). The integer array and the size of the array are passed into the function through parameters. Do not worry about includes. This is only a function, so there is no main routine. The function should fill the array with valid inputs. For invalid input values, inform the user of the error, but do not count that as a valid input.

Q37) Write a function called *Count*. This function is passed a double array along with a parameter that indicates the number of elements in the array. It is also passed a double value. The function computes and returns the number of values in the array that are greater than this double value. Write a complete C function to do this operation. There is no *printf or scanf* in this function. This is only a function, so there is no main routine here!

Q38) Write a function that takes inputs of yards and feet (whole numbers) and calculates and returns an output of the total number of miles (a floating-point value). There are 5280 feet per mile. There are 3 feet in a yard. Use appropriate parameter passing and return mechanisms. Use appropriate datatypes. For example, with inputs of 1760 yards and 1320 ft, the result would be 1.25 miles. Call this function from the main function with different sets of values and print the output.

Q39) write down a complete C program that performs the following

1. Define an array called grades of size 20 and type int
2. Read 20 different values inside the array using scanf. The reading process should be done using loop. The values should be in the range of 0 to 100 inclusive.
3. Calculate the average of the grades.
4. Calculate the highest grade.

Do not use functions in the above program, only the main function.

Q40) write down a complete C program that performs the following

1. Define an array called grades of size 20 and type int, and another array called names to store student names for 20 students (assume maximum length for name is 9 characters).
2. Read 20 different values of grades and names inside the two arrays using scanf. The reading process should be done using loop. The values of grade should be in the range of 0 to 100 inclusive.
3. Calculate the average of the grades.
4. Calculate the highest grade and display the name of the person who has the highest grade.

Q41) Redo 40 but using functions where you have the following functions:

ReadGrades that you pass to it an array of grades and its size.

GetAverage(that you pass to it an array of grades and its size)

GetMax (that you pass to it an array of grades and its size)

Q42) What is the output of the following program?

```
#include<stdio.h>
int main()
{
int a[4][3]= { {3,4,5} ,{31,41,5},{3,2,2}, {1}};
int *p;
for (int i=0;i <4;i++)
{
for (int j=0; j <3;j++)
      printf(" %d ", a[i][j]);

printf("\n");
}
p=&a[0][0];
printf("===========\n");
for (int k=1;k<=5;k++)
      printf(" %d ", *p++);

printf("\n");
printf("===========\n");
p=a[1];
for (int k=1;k<=5;k++)
      printf(" %d ", *p++);


return 0;
}
```

Q43) Consider the Fibonacci series:

1,1,2,3,5,8,13,21,34... Each number, after the second, is the sum of the two numbers before it.

Write down a recursive function fab that computes Fibonacci of the nth number. Note, fab(1) is 1 and fav(2) is 1.

Q44.a) given the following series: 1 2 5 26 677 ….. such that the nth value equals to (n-1 th)^2 +1 and the first value is 1. Write a recursion function named f to compute the nth value. Use for loop to print the values of first 6 values.

Q44.b) Write a C function named MyPower that receives two values v and p, and returns the value of v^p

Note, implement your function using recursion.

Q45) Write a C program that prints below shape of triangle using nested loop

```
*
**
***
****
*****
```

Q46) Write a C program that prints below shape of triangle using nested loop

```
*****
****
***
**
*
```

Q47) Write a C program that prints below shape of rectangle using nested loop

```
*********
*********
*********
*********
*********
```

Q48) Write a C program that prints below shape using nested loop

```
*
**
***
****
*****
*****
****
***
**
*
```

Q49) Write a C program that prints below shape using nested loop

```
    *
   **
  ***
 ****
*****
```

Q50) Write a C program that prints below shape using nested loop

```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Q51) convert the following while loop to an equivalent for loop:

```
#include <stdio.h>
int main()
{
      int x=1;
      int y;
      while (x <=10)
      {
            y=x*x;
            printf("%d %d \n",x,y);
            x+=3;
      }

      return 0;
}
```

Q52)Write a C program that has three functions read_array, sort_array, print_array, read_array will read an array of type integer and size n. Sort will sort the given array and print will print it.

Q53)

What is the output of the following program:

```
#include<stdio.h>

void Print2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
      {
            for (int c=0; c<3;c++)
                  printf("%d ",a[r][c]);
            printf("\n");
      }
}

void Print2DArrayByCols( int  a[][3], int ROWS)
{
      for (int c=0; c<3;c++)
      {
            for ( int r=0; r<ROWS; r++)
                  printf("%d ",a[r][c]);

            printf("\n");
      }
}
void Read2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                  scanf("%d ",&a[r][c]);
}




int Max2DArray( int  a[][3], int ROWS)
```

```
{  int max;
      max=a[0][0];
      for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                  if (a[r][c]>max)
                        max=a[r][c];
      return max;
}


int MaxOfRow( int  a[][3], int ROWS, int searched_row)
{  int max;
      max=a[searched_row][0];
            for (int c=0; c<3;c++)
                  if (a[searched_row][c]>max)
                        max=a[searched_row][c];

      return max;
}

int MaxOfCol( int  a[][3], int ROWS, int searched_col)
{  int max;
      max=a[0][searched_col];
      for ( int r=1; r<ROWS; r++)
                  if (a[r][searched_col]>max)
                        max=a[r][searched_col];
      return max;
}

int SumOfArray (int  a[][3], int ROWS)
{
      int sum=0;
      for ( int r=0; r<ROWS; r++)
            for ( int c=0; c<3;c++)
                  sum+=a[r][c];

      return sum;
}

void SumOfCols (int  a[][3], int ROWS)
{
      int sums[3]={0};
      for ( int r=0; r<ROWS; r++)
            for ( int c=0; c<3;c++)
                  sums[c]+=a[r][c];

      printf("\n sum of cols\n");
      for (int c=0; c<3;c++)
        printf("%d ",sums[c]);

      printf("\n");
}

void SumOfRows (int  a[][3], int ROWS)
{
      int sums[4]={0};
      for ( int c=0; c<3;c++)
```

```
            for ( int r=0; r<ROWS; r++)
                  sums[r]+=a[r][c];

      printf("\n sum of rows\n");
      for (int r=0; r<4;r++)
        printf("%d ",sums[r]);

      printf("\n");

}

bool SearchArray (int  a[][3], int ROWS, int key)
{
      bool found=false;

            for ( int r=0; r<ROWS; r++)
                  for ( int c=0; c<3;c++)
                    if (a[r][c]==key)
                    {
                          printf(" found at location [%d,%d] \n",r,c);
                          return true;
                    }

            return found;

}

bool SearchArrayRow(int  a[][3], int ROWS, int searched_row,int key)
{
      bool found=false;


                  for ( int c=0; c<3;c++)
                    if (a[searched_row][c]==key)
                    {
                          printf(" found at col %d \n",c);
                          return true;
                    }
            return found;
}

bool SearchArrayCol(int  a[][3], int ROWS, int searched_col,int key)
{
      bool found=false;


                  for ( int r=0; r<ROWS;r++)
                    if (a[r][searched_col]==key)
                    {
                          printf(" found at row %d \n",r);
                          return true;
                    }
            return found;
}


int main()
```

```
{
      int b[4][3]={ {1}, {4,5,6} ,{7,8,9}, {1 ,3} };
      printf("================\n");
      Print2DArray(b,4);
      printf("================\n");
      Print2DArrayByCols(b,4);
      printf("================\n");
//    Read2DArray(b,4);
//    Print2DArray(b,4);
      printf("max of array is %d \n",Max2DArray(b,4));
      printf("max of first row is %d \n",MaxOfRow(b,4,0));
      printf("max of third row is %d \n",MaxOfRow(b,4,2));
      printf("max of second col is %d \n",MaxOfCol(b,4,1));
      printf("sum of all Array is %d \n",SumOfArray(b,4));
      printf("================\n");
      SumOfCols(b,4);
      SumOfRows(b,4);
      SearchArray(b,4,8);
      SearchArrayRow(b,4,2,8);
      SearchArrayCol(b,4,1,3);

      return 0;
}
```

Q54)
What is the output from the following statements?

```
      a. if ( 60 <= 12 * 5)
            printf("Hello ");
            printf( " There");

      b. if ('a' > 'b' || 66 > static_cast<int>('A'))
            printf("#*# \n");

      c. if (7 <= 7)
            printf("%d",6 - 9 * 2 / 6)
```

Q55)

2. Suppose that you have the following variable declarations:

```
      int x = 10;
      int y = 15;
      int z = 20;
```

Determine whether the following expressions are true or false.

```
      a. !(x < 10)
      b. x <= 5 || y > 15
      c. (x != 5) && (y == z)
      d. x <= z && (x + y >= z)
```

Q56) What is the output of the following code fragment?

```
int x = 201;
      int y = 101;
      if (x > 100 && y <= 200)
            printf("%d %d %d",x,y ,x + y );
      else
            printf("%d %d %d",x,y ,2* x - y );
```

Q57) What is the output from the following C code fragment?

```
#include <stdio.h>

#include <stdio.h>
int main()
{

      int num = 1;
      while(num < 20)
      {
            if (num% 3==1)
                  printf("num =%d \n",num);
            num += 2;
      }
      printf("the end ");



return 0;
}
```

Q58) What is the output from the following C code fragment if the following values are the inputs to scanf?   10 20 30 40 -1

```
#include <stdio.h>
int main()
{
int sum, num;
      sum = 0;
      scanf("%d",&num);
      while(num != -1)
      {
            sum += num;
            scanf("%d",&num);
      }
      printf("sum =%d \n",sum);

return 0;
}
```

Q59)

How many times will the loop bodies execute in the following loops?

```
a.          int x = 4, y = 50;
      do
      {
            x += 8;

      }while(x < y);
```

```
b.          for (int i=1; i<=5;i+=1)
            for (int j=1;j<=6;j++)
                  for (int k=2;k<=5;k++)
                        printf("hello\n");
c.    int y = 0;
      for(int x = 5; x < 100; x += 5)
      {
            y++;
      }
d.    int x = 10, y = 1000;
      while(x <= y);
      {
            x *= 10;
      }

e.      for (int i=10; i<12;i++)
              for (int j=3;j<7;j++)
                  printf("hello\n");

f.
         for (int i=12; i<20;i+=5)
            for (int j=30;j<37;j+=2)
                  printf("hello\n");
g.
    for (int i=1; i<=3;i+=1)
            for (int j=1;j<=4;j++)
            {                 if (j==3)
                        break;

                  for (int k=2;k<=4;k++)
                        printf("Hello \n");
            }

h. for (int i=1; i<=3;i+=1)
            for (int j=1;j<=4;j++)
            {                 if (j==3)
                        continue;

                  for (int k=2;k<=4;k++)
                        printf("Hello \n");
            }
```

Q60) what is the output of the following code

```
      int a[4][3] = { { 1,2,3} ,{4,5,6} , {7,8,9}, {0}};

      int * p,i;
```

```
printf("==1=====\n");
  p= a[0];
for (i=1;i<=5;i++)
        printf("%d \n",*p++);

  printf("==2=====\n");
  p= a[1];
for (i=1;i<=5;i++)
        printf("%d \n",*p++);

printf("==3=====\n");
  p= &a[2][1];
for (i=1;i<=5;i++)
        printf("%d \n",*p++);
```

Answers:

Q1)

```
1) 10 10 003674D0 003674D0 003674D8
2) 10 10 003674D0 003674D0
3) 100 100 00367038 00367038
4) 300 00367040 00367038
5) 400 00367044
6) 300 300
 loop using pointer
loop: 100 00367038
loop: 200 0036703C
loop: 300 00367040
loop: 400 00367044
loop: 500 00367048
7) 5 5
```

Q2)

1) 500 500

2) 400 400

Q3)

3 5 0 0

5 8 14 9 14

5 5 14 13

7 8 21 36 21

5 7 21 15

Q4)

1) 3 5 0

2) 3 5 13

3) 5 5 13

4) 7 8 15

5) 7 18 36

Q5)

2 4 2 3 0

1 3 5 7 0

4 7 10 13 2

26 7 11 15 0

Q6)

24 6 3 3 0

6 6 3 3 0

47 5 5 5 2

26 3 3 3 0

Q7)

8 0 0 0

2 12 4 0 0

Q8)

5 5 15 20

5 7 22 29 7

Q9)

```
1)  3  5  0  10
```

```
2) 3 5 13 11
3) 5 5 13 11
4) 7 8 15 11
5) 7 18 36 11
```

Q10)

```
================
60 121 182
8 16 24
34 43 52
48 49 50
```

Q11)

```
#include<stdio.h>

void Print2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
      {
            for (int c=0; c<3;c++)
                  printf("%d ",a[r][c]);
            printf("\n");
      }
}

void Read2DArray( int  a[][3], int ROWS)
{     for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                  scanf("%d ",&a[r][c]);
}

void Add2DArrays( int  a[][3], int  b[][3], int  m[][3], int ROWS)
{
      for ( int r=0; r<ROWS; r++)
            for (int c=0; c<3;c++)
                        m[r][c]=a[r][c]+b[r][c];

}

int main()
{
      int x[4][3],y[4][3],z[4][3];
      Read2DArray(x,4);
      Read2DArray(y,4);
      Add2DArrays(x,y,z,4);
      printf("================\n");
      Print2DArray(z,4);

      return 0;
}
```

Q12)

1) 8 8
2) 22 44
3) 8 22

```
Q13A) int Max2DArray( int  a[][5], int ROWS)
{  int max;
      max=a[0][0];
      for ( int r=0; r<ROWS; r++)
            for (int c=0; c<5;c++)
                  if (a[r][c]>max)
                        max=a[r][c];
      return max;
}
```

Q13B)

```
int SumOfArray (int  a[][5], int ROWS)
{
      int sum=0;
      for ( int r=0; r<ROWS; r++)
            for ( int c=0; c<5;c++)
                  sum+=a[r][c];

      return sum;
}
```

Q14) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return true if found false otherwise.

```
bool SearchArray (int  a[][3], int ROWS, int key)
{
      bool found=false;

            for ( int r=0; r<ROWS; r++)
                  for ( int c=0; c<3;c++)
                    if (a[r][c]==key)
                    {
                          printf(" found at location [%d,%d] \n",r,c);
                          return true;
                    }

            return found;

}
```

Q15) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return the row # of location the value was found at if found and return -1 if not found.

```
int SearchArray (int  a[][5], int ROWS, int key)
{
      int res=-1;

            for ( int r=0; r<ROWS; r++)
                  for ( int c=0; c<5;c++)
                    if (a[r][c]==key)
                        return r;


            return res;

}
```

Q16) Write a C function that searches for value key in a a 2D array of size 6 by 5. The function should return the row # and col# of location the value was found at if found and return -1 if not found. The function returns these two valeus using arguments passed by reference.

```
void SearchArray (int  a[][5], int ROWS, int key, int & locR, int &locC)
{
      locR=-1;
      locC=-1;

            for ( int r=0; r<ROWS; r++)
                  for ( int c=0; c<5;c++)
                    if (a[r][c]==key)
                    {
                          LocR=r;
                          locC=c;
                    }
         return; }
```

Q17) Write a C function that computes that maximum of a specific row R in a 2D array of size 6 by 5.

```
int MaxOfRow( int  a[][5], int ROWS, int searched_row)
{  int max;
      max=a[searched_row][0];
            for (int c=0; c<5;c++)
                  if (a[searched_row][c]>max)
                        max=a[searched_row][c];

      return max;
}
```

Q18) Write a C function that computes that total sum of of a specific col C in a 2D array of size 6 by 5.

```
int  SumOfCol (int  a[][5], int ROWS, int col_id)
{
      int sum=0;
      for ( int r=0; r<ROWS; r++)
            sum+=a[r][col_id];
      return sum;}
```

Q19) Write a C function that searches for value key in a specific row R in a 2D array of size 6 by 5. The function should return true if found false otherwise.

```
bool SearchArrayRow(int  a[][3], int ROWS, int searched_row,int key)
{
      bool found=false;


                  for ( int c=0; c<3;c++)
                    if (a[searched_row][c]==key)
                    {
                          printf(" found at col %d \n",c);
                          return true;
                    }
            return found;
}
```

Q20)

PSUT

C Programming

Data

Structure

PSUP

C Programming

DaSa

SPructure

Q21)

PSUT

C Programming

Data

Structure

P

S

D

a

Q22.A)

Determine the value of the following expressions.

a. 'B'
b. 'C'
c. 27.0
d. 9.0

e.1.23

f. 22
g. 34

Q22)

Using the functions in the cmath library, write the following mathematical formulas as C expressions.

a. pow(3.0,2.4)

b.pow(x-y,0.5) or sqrt(x-y)

c. fabs(y - 42.3)

d. (-1*b + sqrt(pow(b,2.0) – 4*a*c)) / 2*a

Q23)

```
bool islower( char ch)

{
   if (ch>= 'a') && (ch<='z')
return true;
else return false;
}
```

Q24)

Write a C function that has three inputs which are integers. The function returns true if the first number raised to the power of the second number equals the third number.

```
bool f1( int a, int b, int c)
{
    if (pow( (float) a, (float) b)==c)
return true;
else return false;
}
```

Q25)

Write a C function that has three inputs which are integers. The function returns true if the first number raised to the power of the second number equals the third number.

```
bool f1( int a, int b, int c)
{
    if (pow( (float) a, (float) b)==c)
return true;
else return false;
}
```

Q26)

```
15  25  30
25  30  40
```

Q27)

a. Declare an array alpha of 17 components of type int.
   int alpha[17];
b. Output the value of the 12th component of the alpha array.
   printf(“%d”,alpha[11]);

c. Set the value of the 5th component of the alpha array to 35.
   alpha[4]=35;
d. Set the value of the 9th component of the alpha array to the sum of the 6th and 13th components of the alpha array.
   alpha[8]= alpha[5]+ alpha[12];
e. Set the value of the 4th component of the alpha array to three times the value of the 8th component minus 57.
   alpha[3]= alpha[7]-57;
f. Output alpha so that five components appear on each line.

```
for (int i=0; i<15;i++)
{ printf(“%d”,alpha[9]);
  If ((i%5)==0)
     printf(“\n”);
```

Q28)

2

7

10

11

16

19

Q29)

a. int grades[10]={};

b. An array of weights has 4 components which have the following values: 5.5, 5.8, 6.3, 6.6.
   float weights []= { 5.5, 5.8, 6.3, 6.6};
c. An array of ages has 6 components which have the values: 80,60,21,42,22,21.
   int ages[]= {80,60,21,42,22,21};
d. An array of symbols which contains the following characters: '$', '%', '@', '!', '|', '&'.

char symbols[]={'$', '%', '@', '!', '|', '&'}

0

Q30) Write a for loop to initialize the following array (int data[10]) with the values 10, 9, 8… 1.

```
int data[10];

for (int i=0;i<10;i++)

      data[i]=10-i;
```

Q31) Given an array of 10 doubles named data, write a loop that loads the array with user input.

```
double data[10];

for (int i=0;i<10;i++)

      scanf("%lf",&data[i]);
```

Q33) Given an array of 100 doubles named data, write a loop that creates the sum of all the array elements.

```
double sum=0;

for (int i=0;i<10;i++)

      sum+=data[i];

printf("%5.2lf",sum);
```

Q34) Write a loop that finds the smallest element in an integer array called data containing 100 elements.

```
double min=data[0];

for (int i=1;i<100;i++)
```

```
        if (data[i]<min)
            min=data[i];
    printf("min is %5.2lf\n",min);
```

Q35) Write a code fragment using ONE for-loop to calculate the sum and product of the elements of a floating point array. The array is declared as shown below. Include any other declarations and initialization statements necessary to perform the operation. Display the sum and product results after the for-loop using *printf*.

```
const int MAX = 15;
float values[MAX] = {98.0, 75.0, 66.5, 78.5, 83.5, 89.0,
  94.0, 43.0, 99.0, 88.0, 42.5, 72.0, 59.5, 77.0, 63.5};
```

```
    float sum=0,prod=1;
    for (int i=0;i<MAX;i++)
    {
        sum+=values[i];
        prod*=values[i];
    }
    printf("sum is %5.2f\n",sum);
    printf("product is %5.2f\n",prod);
```

Q36) Write a function to have a user enter some number of integers into an array. The integer values must be between -100 and +100 inclusive (+100 and -100 should be accepted as valid inputs). The integer array and the size of the array are passed into the function through parameters. Do not worry about includes. This is only a function, so there is no main routine. The function should fill the array with valid inputs. For invalid input values, inform the user of the error, but do not count that as a valid input.

```
void ReadValues( int a[], int n)
{
    int v;
    int count=0;
    while (count <n)
```

```
      {
            scanf("%d",&v);
            if ((v>=-100) && (v <=100))
            { a[count]=v;
              count++;
            }
            else
            {
                  printf(" please enter a value value between -100 and 100\n");
            }
      }
}
```

Q37) Write a function called *Count*. This function is passed a double array along with a parameter that indicates the number of elements in the array. It is also passed a double value. The function computes and returns the number of values in the array that are greater than this double value. Write a complete C function to do this operation. There is no *printf or scanf* in this function. This is only a function, so there is no main routine here!

```
int Count( int a[], int n, int value)
{
      int v;
      int count=0;
    for (int i=0 ; i<n; i++)
            if (a[i]> value)
                  count++;
      return count;
}
```

Q38) Write a function that takes inputs of yards and feet (whole numbers) and calculates and returns an output of the total number of miles (a floating-point value). There are 5280 feet per mile. There are 3 feet in a yard. Use appropriate parameter passing and return mechanisms. Use appropriate datatypes. For example, with inputs of 1760 yards and 1320 ft, the result would be 1.25 miles. Call this function from the main function with different sets of values and print the output.

```
float Miles( int yards, int feet)

{

      return (feet+3*yards)/ 5280.0;

}
```

Q39)

```
#include <stdio.h>
#define SIZE 20
int main()
{
      int grades[SIZE];
      int i,max,sum,temp;
      float avg;

      for ( i=0;i<SIZE;i++)
      {
            do
            {
                  printf("please enter grade in the range of 0 to 100\n");
                  scanf("%d",&temp);
                  if ((temp<0) || (temp>100))
                        printf("please enter a valid grade\n");

            } while ((temp<0) || (temp>100));

            grades[i]=temp;

      }

    sum=0;

      for (i=0;i<SIZE;i++)
            sum+=grades[i];

      avg= sum/(float)SIZE;


      max=grades[0];
      for (i=1;i<SIZE;i++)
      if (grades[i]>max)
```

```
            max=grades[i];

      printf(" Final report\n");
      printf(" ============================\n");
      printf(" average is %.2f \n",avg);
      printf(" heighest grade  is %d \n",max);


return 0;
}
```

Q40)

```
#include <stdio.h>
#define SIZE 20
#define NameLength 9
int main()
{
      int grades[SIZE];
      char names[SIZE][NameLength],ch;

      int i,max,sum,temp,c,index_of_max;
      float avg;

      for ( i=0;i<SIZE;i++)
      {

            // read the name , do not forget to put null character at the
end, also we are
            // using scanf "%c" to cover the case where name might include
space

            c=0;
            printf("please name of student\n");
            while(true)
            {
                  scanf("%c",&ch);
                  if (ch=='\n')
                  {
                        names[i][c]='\0';
                        break;
                  }
                  names[i][c++]=ch;

            }


            // read the grade f
            do
            {
                  printf("please enter grade for student in the range of 0 to
100\n");
                  scanf("%d",&temp);
                  if ((temp<0) || (temp>100))
```

```
                        printf("please enter a valid grade\n");

            } while ((temp<0) || (temp>100));
            grades[i]=temp;

            scanf("%c",&ch); // this is there to consume the enter key that
is in the buffer to allow reading new name




      }

    sum=0;

      for (i=0;i<SIZE;i++)
            sum+=grades[i];

      avg= sum/(float)SIZE;


      max=grades[0];
      index_of_max=0;
      for (i=1;i<SIZE;i++)
      if (grades[i]>max)
      {
            max=grades[i];
            index_of_max=i;
      }

      printf(" Final report\n");
      printf(" ============================\n");
      printf(" average is %.2f \n",avg);
      printf(" heighest grade  is %d \n",max);
      printf(" heighest grade  belongs to the super student %s
\n",names[index_of_max]);


return 0;
}
```

Q41)

```
void ReadValues( int a[], int n)

{

      int v;

      int count=0;

      while (count <n)

      {
        scanf("%d",&v);
        if ((v>=0) && (v <=100))
        { a[count]=v;
          count++;
        }
        else
        {
            printf(" please enter a value value between -100 and 100\n");
        }

    }


}



float GetAverage(int a[], int n )
{
    int sum=0;
    float avg;
    for (int i=0 ; i<n; i++)
        sum+=a[i];

    avg=sum/n;

    return avg;

}
```

```
int GetMax(int a[], int n )

{

      double max=a[0];

      for (int i=1;i<100;i++)

            if (a[i]>max)

                  max=a[i];


      return max;

}
```

Q42)

```
3 4 5

 31 41 5

 3 2 2

 1 0 0

===========

 3 4 5 31 41

===========

 31 41 5 3 2
```

Q43)

```
int fab(int n)
{
      if ((n==1) || (n==2))
            return 1;
      else
            return fab(n-1)+fab(n-2);
}
```

Q44.a)

```
#include <stdio.h>

int f(int n)
{
      if (n==1)
            return 1;
      else
            return f(n-1)*f(n-1)+1;

}
int main()
{
      printf( "%7s %7s \n","  n  ","  f(n)");
      for (int n=1; n<=6;n++)
            printf("%7d%7d\n",n,f(n));


return 0;
}
```

Output

```
    n      f(n)
      1      1
      2      2
      3      5
      4     26
      5    677
      6 458330
```

Q44.b)

```
int MyPower(int v,unsigned int p)
{
      if (p==0)
            return 1;
      else
            return MyPower(v,p-1)*v;

}
```

Q45)

```
#include <stdio.h>

int main()
{
      for ( int i=0; i<5; i++)
      {
            for (int j=0; j<=i; j++)
                  printf("*");
            printf("\n");
      }
      return 0;
}
```

Q46)

```
#include <stdio.h>

int main()
{
      for ( int i=0; i<5; i++)
      {
            for (int j=4; j>=i; j--)
                  printf("*");

            printf("\n");
      }
      return 0;
}
```

Q47)

```
#include <stdio.h>

int main()
{
      for ( int i=0; i<5; i++)
      {
            for (int j=0; j<=8; j++)
                  printf("*");
            printf("\n");
      }
      return 0;
}
```

Q48)

```
#include <stdio.h>

int main()
{
      for ( int i=0; i<5; i++)
      {
         for (int j=0; j<=i; j++)
                  printf("*");
```

```
            printf("\n");
      }
      for ( int i=0; i<5; i++)
      {

            for (int j=4; j>=i; j--)
            printf("*");

            printf("\n");
      }
      return 0;
}
```

Q49)

```
#include <stdio.h>

int main()
{
      for ( int i=0; i<5; i++)
      {
         for (int j=4-i; j>0; j--)
                  printf(" ");

            for (int j=0; j<=i; j++)
                  printf("*");

            printf("\n");

      }


      return 0;
}
```

Q50)

```
#include <stdio.h>

int main()
{
      int range=0;
      for ( int i=1; i<=9; i++)
      {
         for (int j=1; j<=9; j++)
               if ((j>= 5-range) && (j<= 5+range))
                  printf("*");
         else
                printf(" ");

         if (i>=5)
               range-=1;
         else
```

```
             range+=1;

            printf("\n");

      }


      return 0;
}
```

Q51)

```
#include <stdio.h>
int main()
{

      int y;
      for (int x=1;x<=10;x+=3)
      {
            y=x*x;
            printf("%d %d \n",x,y);
      }
      return 0;
}
```

```
Q52) #include <stdio.h>


void print_array( int [], int);
void sort_array( int a[], int x);
void read_array( int a[], int x);

int main()
{
      int A[10];
      read_array(A,10);
      print_array(A,10);
      sort_array(A,10);
      printf(" Array after sort \n");
      print_array(A,10);
      return 0;
}

void read_array( int a[], int x)
{
      for (int i=0;i <x;i++)
            scanf("%d",a[i]);
}
void print_array( int a[], int x)
{
      for (int i=0;i <x;i++)
            printf(" a[%d]=%d \n",i,a[i]);
}
void sort_array( int a[], int x)
{
    int temp;
    for (int i=0;i <x;i++)
        for (int j=0;j<x-1;j++)
            if (a[j] > a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
}
```

Q53)

```
================

1 0 0

4 5 6

7 8 9

1 3 0

================

1 4 7 1

0 5 8 3

0 6 9 0

================

max of array is 9

max of first row is 1

max of third row is 9

max of second col is 8

sum of all Array is 44

================

 sum of cols

13 16 15
```

```
 sum of rows

1 15 24 4

 found at location [2,1]

 found at col 1

 found at row 3
```

Q54)

```
    a. if ( 60 <= 12 * 5)

        cout << "Hello"; /* same as printf in C */

        cout << " There";                   Hello There


    b. if ('a' > 'b' || 66 > static_cast<int>('A'))

        cout << "#*#" << endl;          #*#


    c. if (7 <= 7)

        cout << 6 - 9 * 2 / 6 << endl;          3
```

Q55)

```
int x = 10;

    int y = 15;

    int z = 20;
```

Determine whether the following expressions are true or false.

a. !(x < 10) **true**

b. x <= 5 || y > 15 **false**

c. (x != 5) && (y == z) **false**

d. x <= z && (x + y >= z) **true**

Q56)

201 101 302

Q57)

```
num =1
num =7
num =13
num =19
```

Q58)

100

Q59)

How many times will the loop bodies execute in the following loops?

```
a.     int x = 4, y = 50;
       do
       {
             x += 8;
             printf("Hello %d\n",c++);
       }while(x < y);

6 times

b.           for (int i=1; i<=5;i+=1)
             for (int j=1;j<=6;j++)
                   for (int k=2;k<=5;k++)
                         printf("hello\n");

120 times

c.     int y = 0;
       for(int x = 5; x < 100; x += 5)
       {
             y++;
       }
19 times

d.     int x = 10, y = 1000;
       while(x <= y);
       {
             x *= 10;
```

```
      }
```

0 times, watch for the ;

```
e.      for (int i=10; i<12;i++)
              for (int j=3;j<7;j++)
                  printf("hello\n");
```

8 times

```
f.
         for (int i=12; i<20;i+=5)
            for (int j=30;j<37;j+=2)
                  printf("hello\n");
```

8 times

```
g.
    for (int i=1; i<=3;i+=1)
            for (int j=1;j<=4;j++)
            {                 if (j==3)
                        break;

                  for (int k=2;k<=4;k++)
                        printf("Hello \n");
            }
```

18 times

```
h. for (int i=1; i<=3;i+=1)
            for (int j=1;j<=4;j++)
            {                 if (j==3)
                        continue;

                  for (int k=2;k<=4;k++)
                        printf("Hello \n");
            }
```

27 times.

Q60)

```
==1=====
1
2
3
4
5
==2=====
4
5
```

```
6
7
8
==3=====
8
9
0
0
0
```