Sample Final Exam Questions with Solutions

Q.1 (15 points) Write a **complete C program** for the following algorithm written in pseudocode. Declare all the necessary variables.

Step 1: Read the integer N

Step 2: While (N < 1)

Step 2.1: Print "Error in input, try again!"

Step 2.2: Read the integer N

End_While

Step 3: Result = 1

Step 4: While (N ≥ 1)

Step 4.1: Result = N * Result

Step 4.2: N = N - 1

End_While

Step 5: Print N "! is" Result

Write your C program below:

```
#include <stdio.h>
int main()
{
  int N;
  scanf("%d",&N);

  while (N < 1)
  {
    printf("Error in input, try again!");
    scanf("%d",&N);
  }

  int Result = 1;

  while (N >= 1)
  {
    Result *= N; // or any alternative
    N--;  // or any alternative
  }

  printf("%d ! is %d");
  return 0;
}
```

Q.2. (15 points) Compute the values of the following C expressions assuming that **b, s, e, m** and **k** are integer variables, **z** and **t** are float variable as declared below.

```
int b = 1, s = 3, e = 5, m = 4, k = 10;
float  z = 2.0, t = 5.5;
```

a) `z = b + k * z / m % e - 1` _____0_____

b) `e % s * m > z || k % s * b < t` _____1_____

c) `(b < s) % s+1` _____2_____

d) `++b  &&  s--` _____1_____

e) `m  =  --b ? –e : ++e` _____6_____

Q.3.

a) **(10 points)** Find the output of the following C program and write the output into the given box:

```
#include <stdio.h>
int main()
{ int NUM=0, x=1;
   while(NUM<=8)
   {
     switch (NUM)
     {     case 0:
                 x = 0;
                 printf("%d\n", x);

           case 1:
                 NUM += 1;
                 break;

           case 2:
                 x = 2;
                 printf("%d\n", x);

           case 3:
                 printf("%d\n", x * 2);

           case 4:
                 x = x + 1;
                 NUM=NUM+5;
                 printf("%d\n", x);
                 break;

           default:
                 NUM = NUM * 2;
                 printf("%d\n", NUM);
        }
     }
  return 0;
}
```

```
0

2

4

3

14
```

b) **(8 points)** Consider the given *while loop* below:

```
i = 1;
while ( i <= 10)
{
  printf ("hello\n");
  i=i+2;
}
```

i) Rewrite the code above using a ***for loop***

```
for(i=1; i<=10; i=i+2)

     printf("hello\n");
```

ii) Rewrite the code above using a ***do-while loop***

```
i=1;

do

{ printf("hello\n");

   i=i+2;

 } while (i<=10);
```

Q.4. (18 points)

Complete the missing parts in the following C program to produce sum of the Even numbers in the array and Sum of the Odd numbers in the array. An array of 10 items has to be created and all numbers will be read from keyboard. Sample output should look like as follows:

```
This program reads 10 numbers and displays sum of Even numbers and sum of
Odd numbers.
Enter number : 1
Enter number : 2
Enter number : 3
Enter number : 4
Enter number : 5
Enter number : 6
Enter number : 7
Enter number : 8
Enter number : 9
Enter number : 10
Sum of Even numbers : 30
Sum of Odd numbers : 25
```

```
/* C program to find the sum of Even numbers and Sum of Odd numbers in an
array. */
#include <stdio.h>
int main(void)
{
      int i, sumodd = 0, sumeven = 0;
      int ... a[10].......; /* Array declaration */
      printf("This program reads 10 numbers and displays sum of Even
      numbers and sum of Odd numbers. \n");
      for(...i= 0; i < 10; i++....)
      {    printf(“ ..... "Enter number:".....:”);
           scanf(“ .. %d", &a[i]....);
      }
      for(i= 0; i < 10; i++)
      {     if ( ... a[i]... % 2 == 0)
                .... sumeven += a[i];... ;
            else
                ... sumodd += a[i]...;
       }
       printf(“Sum of Even numbers ... %d\n",sumeven.....);
       printf(“Sum of Odd numbers  ... %d\n",sumodd....);
       return 0;
  }
```

Q.5. (16 points) Write a **C function** that models the following mathematical function. For example, $f(-3.2)$ returns -3.2 + 2 = -1.2. Assume that **x** is a float type variable.

$$f(x) = \begin{cases} x + 2\,, & x < -1 \\ x^2\text{-}, & -1 \le x \le 1 \\ -x + 2, & x > 1 \end{cases}$$

```
float f(float x)
{
  if (x < -1)

    return x + 2;

  else if ((x >= -1) && (x <= 1))

    return x * x;

  else if (x > 1) // <--- optional

    return –x + 2
}
```

Q.6. (20 points) Draw a flowchart and then write a C program to find the count of numbers which are fully divisible by 5 for a set of integer numbers. When the user enters 0, the program must display the computed count. A sample output of the program is given as follows:

```
This program finds count of numbers which are fully divisible by 5.
Enter integers (0 to terminate): 8 25 71 5 10 0
The count is: 3
```

```
#include <stdio.h>
int main()
{
   int number, count=0;
   printf("This program finds
    count of numbers which are
    fully divisible by 5.\n");
    printf("Enter integers (0
     to terminate): ");

    scanf("%d", &number);
    while(number != 0)
    {
          if (number%5==0)
               count++;
          scanf("%d", &number);
    }
    printf("The count is: %d\n",
     count);
    return 0;
}
```

Flowchart: