Indian Institute of Technology, Kharagpur

Department of Computer Science and Engineering

Class Test 2, Autumn 2016-17

Programming and Data Structure (CS 11001 / CS 10001)

Students: 681 **Date**: 15-Mar-17

Full marks: 20 **Time**: 7:00pm–8:00pm

Answer the questions in the spaces provided on the question sheets. You may use the last page of this booklet for your rough work. No other supplementary sheets will be given to you.

Roll Number		Section	
Name			

Question:	1	2	3	Total
Points:	4	10	7	21
Score:				

1. Assume that we use 2's complement notation to represent signed integers in 8 bits.

   (a) (2 points) Find the 8 bit representation of the following numbers:

   i. 14

0	0	0	0	1	1	1	0

   ii. -19

1	1	1	0	1	1	0	1

   (b) (2 points) Find the result when you add 14 with -19 in 2's complement form. Write the 2's complement result and then convert it to its decimal equivalent.

   2's complement result

1	1	1	1	1	0	1	1

   Equivalent decimal number: -5

2. What will be printed when the following programs execute?

   (a) (2 points)

```
int main ( ) {
     int a[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} ;
     int i, k=5, n=9;
     for (i=k; i<n; i++)
          a[i+1] = a[i] ;
     for (i=0; i<n; i++)
          printf ("%d, ", a[i]);
    return 0;
}
```

Solution: 0 1 2 3 4 5 5 5 5

(b) (2 points)

```
#include <stdio.h>
#define N 8
int main()
{
    int a[N] = { 4, 2, 8, 6, 5, 3, 7, 1 };
    int i, j, swap;
    for (i = 1; i < N; i++) {
        for (j = i; j > 0; j--) {
            if (a[j] < a[j-1]) {
                swap = a[j-1];
                a[j-1] = a[j];
                a[j] = swap;
            }
        }
    }
    for (j = 0; j < N; j++)
             printf("%d ", a[j]);
    printf("\n");
    return 0;
}
```

Solution: 1 2 3 4 5 6 7 8

(c) (3 points)

```
#include <stdio.h>
int ternarySearch(int arr[], int l, int r, int x)
{
  printf("L: %d R: %d\n",l,r );
   if (r >= l)
   {
        int mid1 = l + (r - l)/3;
        int mid2 = mid1 + (r - l)/3;
        if (arr[mid1] == x)  return mid1;
        if (arr[mid2] == x)  return mid2;
        if (arr[mid1] > x) return ternarySearch(arr, l, mid1-1, x);
        if (arr[mid2] < x) return ternarySearch(arr, mid2+1, r, x);
        return ternarySearch(arr, mid1+1, mid2-1, x);
   }
   return -1;
}
int main()
{
  int a[] = {1,2,3,4,5,6,7,8};
  ternarySearch(a,0,7,4);
}
```

Solution: L: 0 R: 7
L: 3 R: 3

(d) (3 points)

```
#include <stdio.h>
int main()
{
    int i,a[] = { 7, 12, 1, -2, 0, 15, 4, 11, 9};
    quickSort( a, 0, 8);
    return 0;
}
void quickSort( int a[], int l, int r)
{
    int j;
    if( l < r ) {
        j = partition( a, l, r);
        quickSort( a, l, j-1);
        quickSort( a, j+1, r);
}
}
int partition( int a[], int l, int r)
{
    int pivot, i, j, t;
    pivot = a[l];
    printf("Pivot: %d\n",pivot);
    i = l;
    j = r+1;
    while( 1) {
        do { ++i; } while(a[i]<=pivot && i<=r);
        do { --j; } while( a[j] > pivot );
        if( i >= j ) break;
        t = a[i];  a[i] = a[j];  a[j] = t;
    }
    t = a[l]; a[l] = a[j]; a[j] = t;
    return j;
}
```

Solution: Pivot: 7
Pivot: 0
Pivot: 1
Pivot: 15
Pivot: 9
Pivot: 12

3. (a) (2 points) Define a structure `struct point` to represent a point in the 2-dimensional coordinate system.

Solution:

```
struct point {
float x,y;
};
```

(b) (2 points) Write a function that takes a parameter of type `struct point` as input, and returns the Euclidean distance of the point from the origin, i.e., $(0, 0)$.

Solution:

```
float dist(struct point p)
{
```

```
   return sqrt{p.x*p.x + p.y*p.y};
}
```

(c) (3 points) Write another function that takes an array of points and the size of the array as input, and returns the point which has the smallest Euclidean distance from origin $(0, 0)$.

Solution:

```
struct point closest(struct point pts[], int n)
{
  int i,mini=0;
  float min;
  min=dist(pts[0]);
  for(i=1;i<n;i++) {
    if(dist(pts[i]) < min) {
      mini = i;
      min = dist(pts[i]);
    }
  }
  return pts[mini];
}
```