



Clearly print (not sign) your name:

You may use one page of hand written notes (both sides) and a dictionary. You may not use any other books. You may not use phones, calculators, nor other type of non-organic computer. Write answers on exam. You may use extra paper which you may turn in or keep, as you see fit. Time Limit: 75 minutes.

1) Bit Operators: This C program compiles and runs. What is its output?

```
1) #include <stdio.h>
2) void main(void)
3) { unsigned char x = 103;
4)
5)     unsigned char a = x >> 2;
6)     unsigned char b = x & 16;
7)     unsigned char c = x & 15;
8)     unsigned char d = x | 32;
9)     unsigned char e = x | 15;
10)    unsigned char f = x ^ 15;
11)
12)    printf("a=%d, b=%d, c=%d, d=%d, e=%d, f=%d\n",
13)           a, b, c, d, e, f);
14) }
```



2) **Bubble Sort.** This C program compiles and runs. What is its output?

```
1) #include <stdio.h>
2)
3) void printArray(int array[], int n)
4) {
5)     int i;
6)     printf("{");           // <-----
7)     for (i=0; i<n; i++)
8)     {
9)         printf("%d ", array[i]); // <-----
10)    }
11)    printf("}\n");         // <-----
12) }
13)
14) void bubbleSort(int array[], int n)
15) {
16)     int i, swap = 1;
17)     while(swap)
18)     {
19)         printArray(array, n);
20)         swap = 0;
21)         n--;
22)         for (i=0; i<n; i++)
23)         {
24)             if (array[i] > array[i+1])
25)             {
26)                 printf("  swap: %d <--> %d\n", array[i], array[i+1]); //<-
27)                 int tmp = array[i];
28)                 array[i] = array[i+1];
29)                 array[i+1] = tmp;
30)                 swap = 1;
31)             }
32)         }
33)     }
34) }
35)
36)
37) void main(void)
38) {
39)     int numList[] = {9, 3, 1, 7, 5, 2, 8};
40)     int n = sizeof(numList)/sizeof(int);
41)     bubbleSort(numList, n);
42)     printf("Return from bubbleSort()\n"); // <-
43)     printArray(numList, n);
44) }
```



3) This C program compiles and runs. What is its output?

```
1) #include <stdio.h>
2) void main(void)
3) {
4)   char bitList[] = "00000000"; //8 bits
5)   char* bit = bitList;
6)   unsigned char n=99; //same number from question 1
7)   unsigned char powerOf2 = 128;
8)
9)   while(*bit)
10)  {
11)    printf("n=%3d, powerOf2=%d\n", n, powerOf2); //<--
12)    if (n >= powerOf2)
13)      {
14)        *bit = '1';
15)        n = n-powerOf2;
16)      }
17)    bit++;
18)    powerOf2 = powerOf2 >> 1;
19)  }
20)  printf("%s\n", bitList); //<--
21) }
```



4) This C program compiles and runs. What is its output?

```
1) #include <stdio.h>
2) #include <string.h>
3)
4) char *findSubstring(char *str, char *needle)
5) {
6)     int len = strlen(needle);
7)     int n = 0;
8)
9)     while (*str)
10)    {
11)        if ( *str == *(needle+n))
12)        {
13)            printf("Match: %c, n=%d\n", *str, n);    //<--
14)            n++;
15)            if (n == len) return (str-len)+1;
16)        }
17)        else
18)        {
19)            str -= n;
20)            n = 0;
21)        }
22)        str++;
23)        printf("  %c%c\n", *str, *(needle+n));    //<--
24)    }
25)    return NULL;
26) }
27)
28)
29) void main(void)
30) {
31)     char* sentence = "ThingThereIsScary";
32)     char* word = "There";
33)     char* result = findSubstring(sentence, word);
34)     printf("result=%s\n", result);    //<--
35) }
```



5) **Quicksort:** This C program compiles and runs. What is its output?

```
1) #include <stdio.h>
2) void swap(int v[], int i, int j)
3) {
4)     int c = v[i];
5)     v[i] = v[j];
6)     v[j] = c;
7) }
8)
9) void quicksort(int v[], int left, int right)
10) { int i, last;
11)     printf("[%d, %d]\n", left, right);    //<-----
12)     if (left >= right) return;
13)
14)     swap(v, left, (left+right)/2);
15)     last = left;
16)     for (i=left+1; i <= right; i++)
17)     {
18)         if (v[i] < v[left])
19)             { last++;
20)               swap(v, last, i);
21)             }
22)     }
23)
24)     swap(v, left, last);
25)     quicksort(v, left, last-1);
26)     quicksort(v, last+1, right);
27) }
28)
29)
30) void main(void)
31) {
32)     int v[] = {11, 77, 88, 22, 55, 33};
33)
34)     int arraySize = sizeof(v)/sizeof(int);
35)     quicksort(v, 0, arraySize-1);
36) }
```