1. What is the output from the program shown below?

```
#include <stdio.h>
int function(int data[], int z) {
    if (z == 1) {
        if (data[0] > 0) return 1;
        else             return 0;
    }
    if (data[z-1] > 0)
        return 1 + function(data, z-1);
    else
        return function(data, z-1);
}
int main(void) {
    int a[3] = { -1, 0, 1 };
    int b[4] = { 10, 5, 0, -5 };
    int c[5] = { 1, 2, 3, 4, 5 };
    int d[4] = { 11, -22, 33, -44 };
    int e[1] = { -1 };
    printf("%d\n", function(a, 3) );
    printf("%d\n", function(b, 4) );
    printf("%d\n", function(c, 5) );
    printf("%d\n", function(d, 4) );
    printf("%d\n", function(e, 1) );
    return 0;
}
```

2. What is the output from the program shown below?

```
#include <stdio.h>
void function1(char *str, int num) {
        int a = 0;
        while ( a < num%10 ) {
                printf("%c", str[a]);
                a = a + num/10;
        }
        printf("\n");
        return;
}
void function2(int *x, int *y, int *z) {
        int temp = *x;
        *x = *x + *y + *z;
        *y = temp + *y;
        *z = temp - *z;
        return;
}
int main(void) {
        int a=27, b=39, c=13;
        function1("Roll-Tide", a);
        function1("ComputerScience", b);
        function1("CS100", c);
        function2(&a, &b, &c);
        printf("%d and %d and %d\n", a, b, c);
        function2(&a, &b, &c);
        printf("%d and %d and %d\n", a, b, c);

        return 0;
}
```

3. What is the output from the program shown below when run with the input **2 4 3 6 9 0** ?

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
        int data;
        struct node *next;
} Node;

Node *function1(Node *ptr, int num) {
        Node *new1 = malloc( sizeof(Node) );
        new1->data = num;
        Node *new2 = malloc( sizeof(Node) );
        new2->data = num;
        if (ptr == NULL || num%2 == 0) {
                new1->next = new2;
                new2->next = ptr;
                return new1;
        }
        Node *temp = ptr;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = new1;
        new1->next = new2;
        new2->next = NULL;
        return ptr;
}

void function2(Node *ptr) {
        printf("[ ");
        for (Node *temp = ptr; temp != NULL; temp = temp->next)
                printf("%d ", temp->data);
        printf("]\n");
        return;
}

int main(void) {
        Node *ptr = NULL;
        int value;
        scanf("%d", &value);
        while ( value != 0 ) {
                ptr = function1(ptr, value);
                function2(ptr);
                scanf("%d", &value);
        }
        return 0;
}
```