

## SCC 120 Introduction to Data Structures

### Workshop One : Simple Arrays, Flow Charts and Playing Computer

Q1: What is the decimal equivalent of the 2's complement 8-bit binary 11110010?

Q2: What is the 2's complement 8-bit binary equivalent of the decimal number -20?

Q3:

Here is a 'C' program :

---

```
#include <stdio.h>

int search(int nums[], int size, int target)
{
    int position = -1;
    for (int i = 0; position == -1 && i < size; i++)
    {
        if (nums[i] == target)
            position = i;
    };
    return position;
} // end of function "search"

int sumUp(int nums[])
{
    int count = 0;
    int index = 0;
    while (nums[index] >= 0)
    {
        count = count + nums[index];
        index++;
    }
    return count;
} // end of function "sumUp"

void tutorial_1a()
{
    int numbers[] = {14, 21, 3, 10, 8, 40, -1};
    int sum = sumUp(numbers);
    printf("sum = %d\n",sum);

    int pos = search(numbers,6,3);
    printf("looking for 3. pos = %d ",pos);
    if (pos != -1) printf(" value in array = %d\n",numbers[pos]); else
    printf("target not found\n");

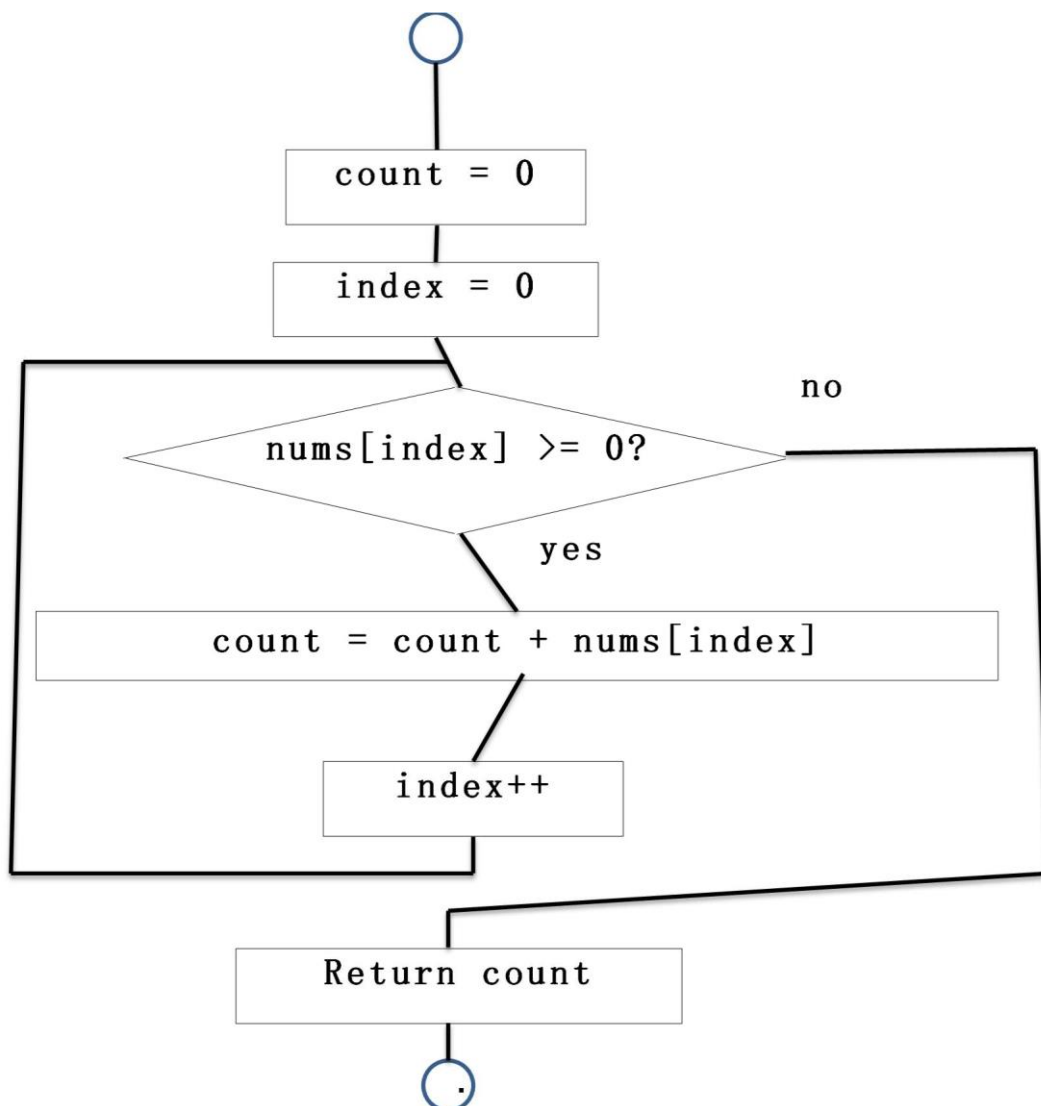
    pos = search(numbers,6,9);
    printf("looking for 6. pos = %d ",pos);
    if (pos != -1) printf(" value in array = %d\n",numbers[pos]); else
    printf("target not found\n");
} // end of procedure "tutorial 1a"
```

---

#### Exercise One : the sumUp function

This function takes as a single parameter an integer array containing a set of numbers which are terminated by a negative number. You can see the actual parameter and how it is set up in the calling procedure "tutorial\_1a". The "sumUp" function adds together all the numbers in the array, stopping when it reaches the terminating negative number.

Here is a flowchart for the sumUp function.



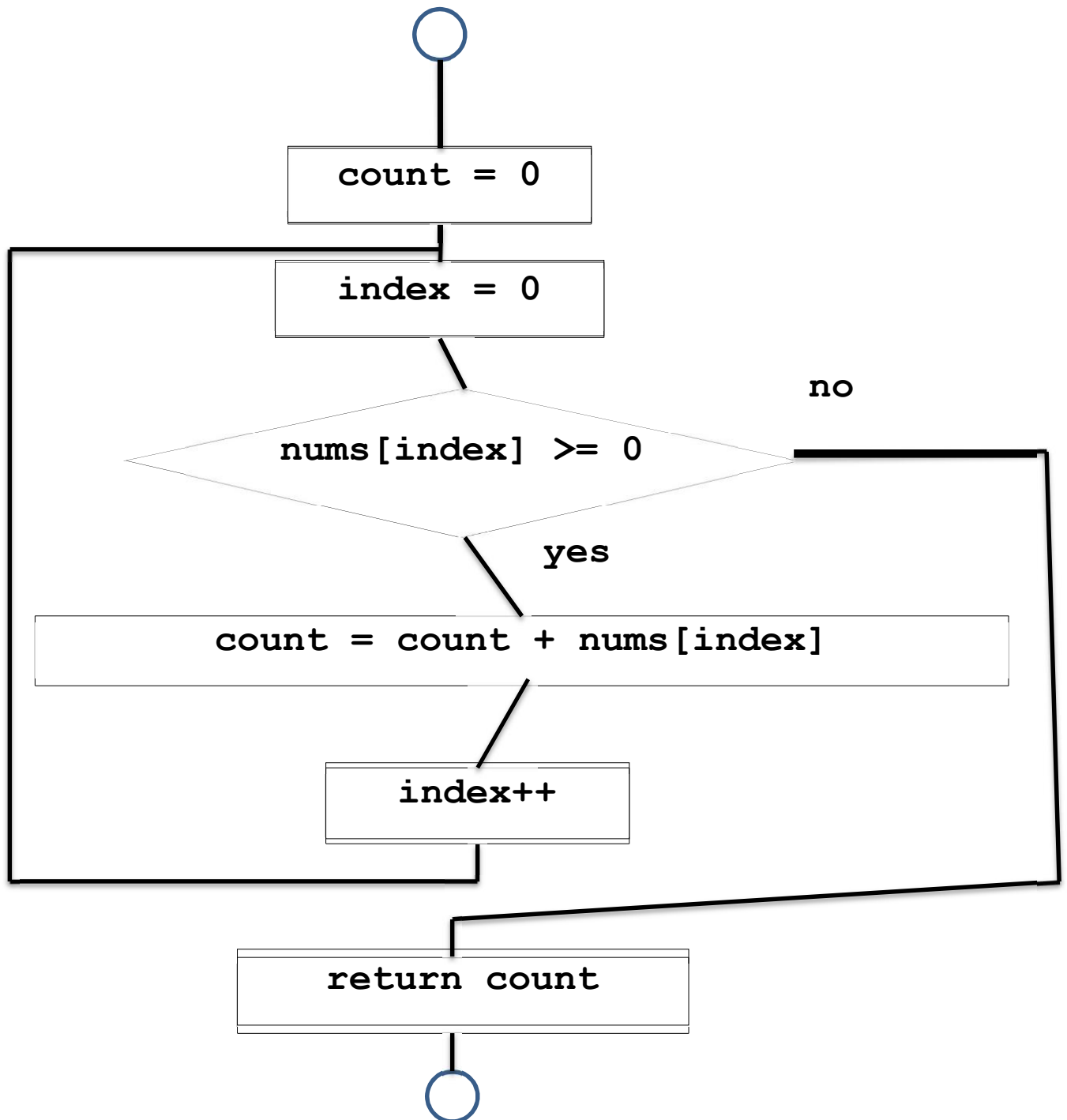
Play computer with this function, with the data given in the code, using the form on the next page.

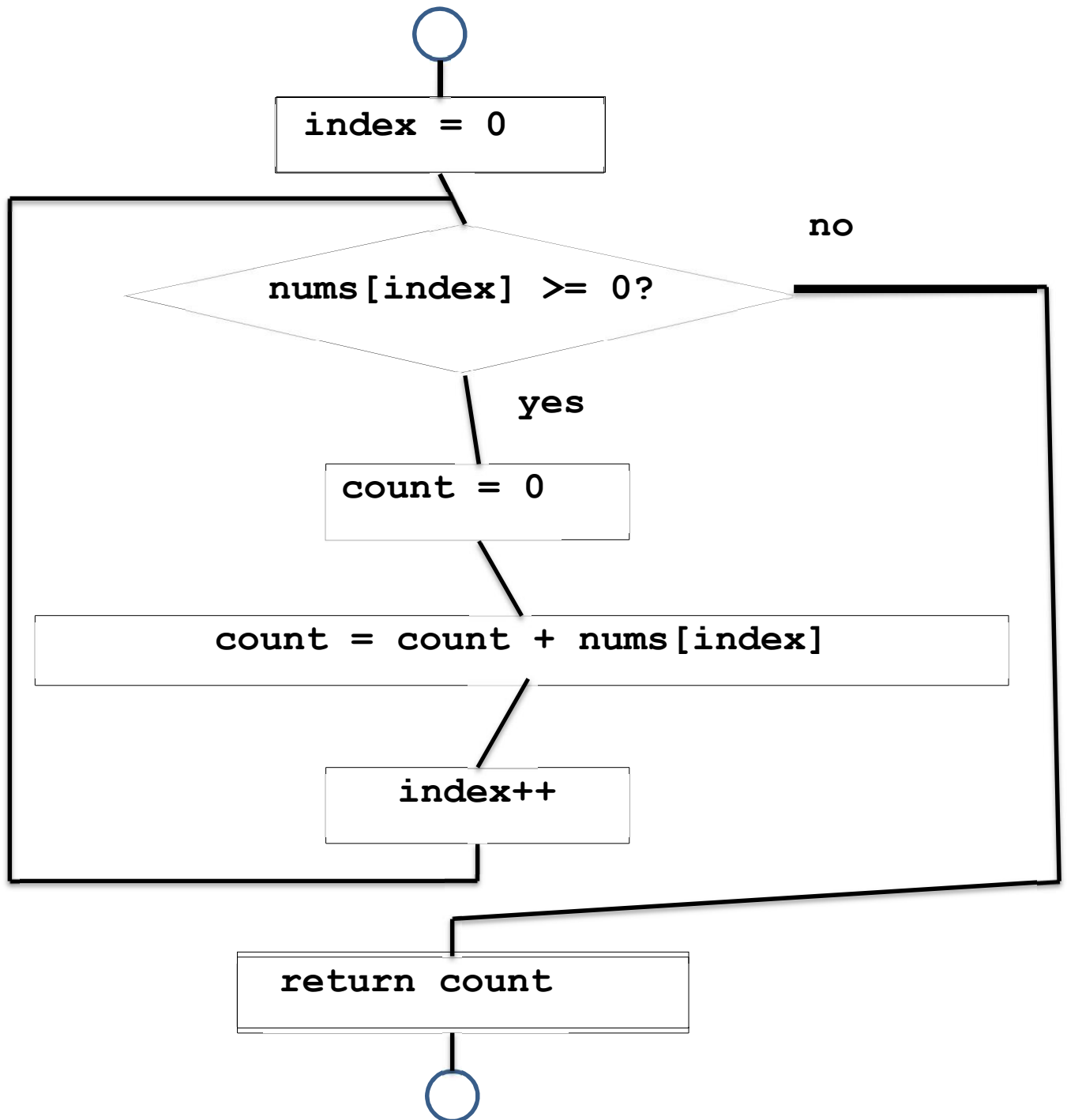
	0	1	2	3	4	5	6
nums	14	21	3	10	8	40	-1

count	index	nums[index]	nums[index] >=0?

The count should end up with the value 96.

What would happen if the programmer made some mistakes and their flowchart looked like these next two?





Play computers with these if you can't understand why they are wrong. Getting things in the right sequence is crucial!

**Exercise Two** : Draw a flowchart for the “*search*” function.

The signature for this function has three parameters : the array of integers to be searched; the “size” of the array – really, the number of elements to be searched; and finally, the target value we are looking for. The function either returns the position of the target value, if it is present, or -1 to indicate that the target value is not present.