

Can you solve this? What two positive numbers can make this possible: $n1 > n1 + n2$ OR $n2 > n1 + n2$
Enter them here nc saturn.picoctf.net 54967.

```
#include <stdio.h>

#include <stdlib.h>

static int addIntOvf(int result, int a, int b) {

    result = a + b;

    if(a > 0 && b > 0 && result < 0)

        return -1;

    if(a < 0 && b < 0 && result > 0)

        return -1;

    return 0;

}

int main() {

    int num1, num2, sum;

    FILE *flag;

    char c;

    printf("n1 > n1 + n2 OR n2 > n1 + n2 \n");

    fflush(stdout);

    printf("What two positive numbers can make this possible: \n");

    fflush(stdout);

    if (scanf("%d", &num1) && scanf("%d", &num2)) {

        printf("You entered %d and %d\n", num1, num2);

        fflush(stdout);

        sum = num1 + num2;

        if (addIntOvf(sum, num1, num2) == 0) {

            printf("No overflow\n");

            fflush(stdout);

            exit(0);

        } else if (addIntOvf(sum, num1, num2) == -1) {

            printf("You have an integer overflow\n");

            fflush(stdout);

        }

    }

    if (num1 > 0 || num2 > 0) {

        flag = fopen("flag.txt", "r");

        if(flag == NULL){
```

```

printf("flag not found: please run this on the server\n");

fflush(stdout);

exit(0);

}

char buf[60];

fgets(buf, 59, flag);

printf("YOUR FLAG IS: %s\n", buf);

fflush(stdout);

exit(0);

}

}

return 0;

}

```

So here we have Integer overflow **Integer overflow** occurs when a calculation or data conversion creates a number **outside the range** that can be represented by a given integer type (like int, long, etc.). When this happens, the value **wraps around**—meaning it loops back to the smallest representable number or vice versa. `int x = 2147483647;`

```
x = x + 1;
```

Then x becomes -2147483648 — that's overflow.

So here we are doing the same Maximum signed integer value is 2,147,483,647.

We are providing 2147483647 and 1

The screenshot shows the picoCTF website interface. The main content area displays the 'two-sum' challenge, which is a Medium difficulty Binary Exploitation problem. The description asks the user to find two positive numbers that can make a given sum possible. A terminal window shows the user entering the numbers 2147483647 and 1, which triggers an integer overflow. The terminal output shows the flag: `YOUR FLAG IS: picoCTF{Tw0_Sum_Integer_Bu773R_0v3rflow_ccd078bd}`. The challenge has been solved by 8,222 users and has an 87% like rating. A 'Submit Flag' button is visible at the bottom of the challenge page.