

# 6th Marathon of Parallel Programming

## SBAC-PAD'11

*October 26<sup>th</sup>, 2011.*

### **Warmup Rules**

For all problems, read carefully the input and output session. For all problems, a sequential implementation is given, and it is against the output of those implementations that the output of your programs will be compared to decide if your implementation is correct. You can modify the program in any way you see fit, except when the problem description states otherwise. You must upload a compressed file with your source code, the *Makefile* and an execution script. The program to execute should have the name of the problem. You can submit as many solutions to a problem as you want. Only the last submission will be considered. The *Makefile* must have the rule *all*, which will be used to compile your source code before submit. The execution script must follow the *Cluster Enterprise3* submitting rules – it will be inspected not to corrupt the target machine.

All teams have access to the target machine during the marathon. Your execution is queued by the Sun Grid Engine (SGE) and does not have concurrent process. During the marathon, SGE will be examined to find any suspect execution.

The execution time of your program will be measured running it with *time* program and taking the real CPU time given. Each program will be executed at least twice with the same input and only the smaller time will be taken into account. The sequential program given will be measured the same way. You will earn points in each problem, corresponding to the division of the sequential time by the time of your program (*speedup*). The team with the most points at the end of the marathon will be declared the winner.

*This problem set contains 1 problem; pages are numbered from 1 to 2.*

# Problem A

## Fibonacci numbers

Fibonacci numbers is a well-known sequence of numbers that follows the recurrence relation  $F_n = F_{n-1} + F_{n-2}$ , with seeds values  $F_0=0$  and  $F_1=1$ .

The Fibonacci numbers are used in many researches such as mathematics, computer science and physics. Besides, it can be found in nature too, such as branching in trees, arrangement of leaves on a stem, the fruitlets of a pineapple among others.

Write a program that calculates the Fibonacci sequence.

### Input

The input contains only one test case. The first line contains only one number ( $N$ ) that will be used to search the  $F_n$  number on the Fibonacci sequence ( $0 \leq N \leq 10^5$ ).

*The input must be read from standard input.*

### Output

For the input test case, your program will output one line containing the number of the Fibonacci sequence.

*The output must be written to standard output.*

### Example 1

Input	Output for the input
11	89

### Example 2

Input	Output for the input
20	6765

```

#define MAX 100010
#define LEN 25001
char seq[MAX][LEN];
void add(int a, int b) {
    int i, aux, s;
    for (i = 0; aux = 0; seq[a][i] != '\0' && seq[b][i] != '\0';
        i++) {
        s = seq[a][i] + seq[b][i] + aux - '0' - '0';
        aux = s / 10;
        seq[a + 1][i] = s % 10 + '0';
    }
    while (seq[a][i] != '\0') {
        s = seq[a][i] + aux - '0';
        aux = s / 10;
        seq[a + 1][i] = s % 10 + '0';
        i++;
    }
    while (seq[b][i] != '\0') {
        s = seq[b][i] + aux - '0';
        aux = s / 10;
        seq[a + 1][i] = s % 10 + '0';
        i++;
    }
    if (aux != 0)
        seq[a + 1][i++] = aux + '0';
    seq[a + 1][i] = '\0';
}
int main() {
    int n, i, len;
    seq[0][0] = '0';
    seq[0][1] = '\0';
    seq[1][0] = '1';
    seq[1][1] = '\0';
    for (i = 2; i < MAX; i++)
        add(i - 1, i - 2);
}

scanf("%d", &n);
len = strlen(seq[n]);
for (i = 0; i <= len - 1; i++)
    printf("%c", seq[n][len - 1 - i]);
fflush(stdout);
return EXIT_SUCCESS;

```