Simboliskie aprēķini

Aprēķiniet **a^b**, ņemot vērā, ka rezultāts var būt lielāks par 2^{64}

3^{100} = 515 377 520 732 011 331 036 461 129 765 621 272 702 107 522 001

```
//~~~ basic type for storing digits of a number ~~~~~~~~~~~~~~~~~~~~~~~~~~

typedef int digits[1000]; // 1024 -> 4, 2, 0, 1, -1 ...

int length(digits dig){
    int len = 0;
    while (dig[len] >= 0) len++;
    return len;
}

string toString(digits dig) {
    string res{ "" };
    for (int d = length(dig) - 1; d >= 0; d--) {
        res += dig[d] + '0';
        if (d % 3 == 0 && d != 0) res += ' ';
    }
    return res;
}
```

Simboliskie aprēķini

```
//~~~ mathematical operations ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

void mul(digits num, int multiplier) {
    int carry = 0, d;
    for (d = 0; num[d] >= 0; d++) {
        num[d] = num[d] * multiplier + carry;
        carry = num[d] / 10;
        num[d] %= 10;
    }
    while (carry) {
        num[d++] = carry % 10;
        carry /= 10;
    }
    num[d] = -1;
}

void power(digits result, int base, int exponent) {
    result[0] = 1;
    result[1] = -1;
    for (int i = 0; i < exponent; i++) mul(result, base);
}
```

Simboliskie aprēķini

```
#include <iostream>
#include <string>

using namespace std;

typedef int digits[1000]; // 1024 -> 4, 2, 0, 1, -1 ...

int length(digits dig) {…}
string toString(digits dig) {…}
void mul(digits num, int multiplier) {…}
void power(digits result, int base, int exponent) {…}

int main()
{
    digits pow;
    int base, exp;
    cout << "Enter the base of power: "; cin >> base;
    cout << "Enter the exponent of power: "; cin >> exp;
    power(pow, base, exp);
    cout << base << " ^ " << exp << " = " << toString(pow) << "\n";
}
```

Mājas uzdevums: simboliskie aprēķini

Pieprasīt no klaviatūras veselo skaitli **n** (**0 < n < 10000**), izmantojot simboliskus aprēķinus, aprēķināt un izvadīt uz ekrāna:

A: n! (n! = 1·2·3·4·…·n)

30! = 265 252 859 812 191 058 636 308 480 000 000

50! = 30 414 093 201 713 378 043 612 608 166 064 768 844 377 641 568 960 512 000 000 000 000

B: $3^n + 2^n$

$3^{100}+2^{100}$ = 515 377 520 732 011 332 304 111 729 993 850 674 198 810 727 377

$3^{200}+2^{200}$ = 265 613 988 875 874 769 338 781 322 035 779 628 436 171 496 912 384 771 516 537 054 080 255 093 423 505 176 777 177 534 345 377

C: $3^n - 2^n$

$3^{100}-2^{100}$ = 515 377 520 732 011 329 768 810 529 537 391 871 205 404 316 625

$3^{200}-2^{200}$ = 265 613 988 875 874 769 338 781 322 035 779 625 222 295 408 394 404 220 432 612 869 397 929 888 379 099 189 211 591 863 742 625

D: $3^n \times 2^n$

$3^{50} \times 2^{50}$ = 808 281 277 464 764 060 643 139 600 456 536 293 376

$3^{100} \times 2^{100}$ = 653 318 623 500 070 906 096 690 267 158 057 820 537 143 710 472 954 871 543 071 966 369 497 141 477 376