LAB 0319

1. Please write a subroutine void fixsumNum(int x[], int n) which find out how many equal prefix sums and suffix sums are there in an given array x[] with positive elements. And please printout these pairs in above subroutine. The prefix sums in an array x[] with n elements is x[0]+ x[1]+ x[2]+ x[3]+...+x[i], and the suffix sums in the same array is x[j]+ x[j+1]+ x[j+2]+ x[j+3]+...+x[n-1].
   You can straightly create an array with 10 elements
   Ex: x[10] = {1,4,5,2,8,3,4,3,5,5}
   1+4 (prefix sum) = 5 (suffix sum)
   1+4+5 (prefix sum) = 5+5 (suffix sum)
   1+4+5+2+8 = 5+5+3+4+3
   1+4+5+2+8+3+4+3 = 5+5+3+4+3+8+2
   1+4+5+2+8+3+4+3+5 = 5+5+3+4+3+8+2+5
   1+4+5+2+8+3+4+3+5+5 = 5+5+3+4+3+8+2+5+4+1
   There are 6 pairs.

2. **Please use at least 2 methods of vector<>, ex: push_back(), size(), back(), etc. You can also use vector as array form.**

   A golden rectangle is a rectangle satisfying (long side) / (short side) = phi(1.618…). Artists may want to construct a 2D array with row / column = phi, so they can fill colors on each pixel and finally finish an image with golden ratio(phi). But it's impossible since phi is an irrational number.

   However, with the help of Fibonacci sequence we can almost represent phi as the division of two positive integer.

$$F_n = F_{n-1} + F_{n-2}$$

$$F_0 = 0, F_1 = 1$$

$$\frac{F_n}{F_{n-1}} \approx phi$$

   This shows that as n becomes larger and larger, the division converges to phi.

   Given an absolute error err as input, please write a function

   vector<int> fib_sequence(double err)

   that returns a vector consisting of the Fibonacci sequence from first element(F_0) to F_n where $\frac{F_n}{F_{n-1}} - 1.618 < err$

The function has
Input : A double parameter err
Output : A vector representing Fibonacci sequence {F_0, F_1, F_2…, F_n}

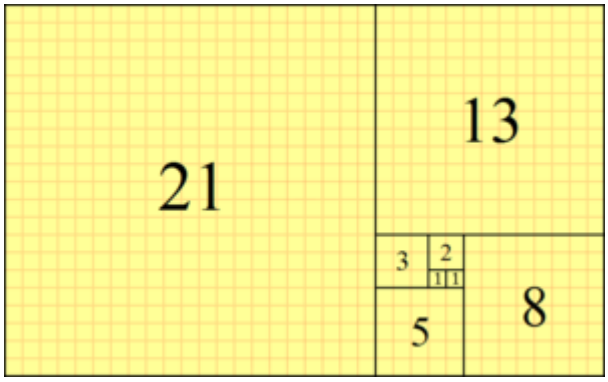
You can use following code for help

```
int main()
{

   double err;
   vector<int> fibs;

   printf("Please input the absolute error : ");
   scanf("%lf", &err);

   fibs = fib_sequence(err);

   for(int i=0; …… ; i++)
      printf("%d ", fibs[i]);

   return 0;
}
```

Demo should be like following:

```
[u0310742@ce37 ~/lab02_0319]$ ./2
Please input the absolute error : 0.005
0 1 1 2 3 5 8 13 21
```

3. Fibonacci sequence can also be used to construct a larger and more accurate golden rectangle by combining the small ones.



Using the result you generate from question 2

Please write a function

int** fibSquare(vector<int> fibs, int h, int w)

which take 3 parameter as input,

fibs is the Fibonacci sequence generated from question 2

h, w represents the height and width of the square

and output a h*w 2D array representing a rectangle as the picture shows. The values in the array should be corresponding to the numbers in Fibonacci sequence, for example:

$$\text{arr}[0-20][0-20] = 21$$
$$\text{arr}[21-33][0-12] = 13$$
$$\dots$$

Note : There are more than one ways to construct your rectangle, but you only need to show one.

Note : You can also return 2D vector instead of array, namely

vector< vector <int> > fibSquare(vector<int> fibs, int h, int w)

You can use following code for help

```
int main()
{
   //same as question 2
   vector<int> fibs;
   double err;

   printf("Please input the absolute error \n");
   scanf("%lf", &err);

   fibs = fib_sequence(err);

   //array size h*w are relative with fibs
   int h = ……
   int w = ……

   int **result = fibSquare(fibs, h, w);

   for(int i=0;i<h;i++){
      for(int j=0;j<w;j++)
            printf("%2d", result[i][j]);
      printf("\n");
   }
}
```

Output should be like following :

```
[u0310742@ce37 ~/lab02_0319]$ ./3
Please input the absolute error
0.005
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121212113131313131313131313131313
21212121212121212121212121212121212121 3 3 3 2 2 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 3 3 3 2 2 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 3 3 3 1 1 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 5 5 5 5 5 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 5 5 5 5 5 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 5 5 5 5 5 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 5 5 5 5 5 8 8 8 8 8 8 8 8
21212121212121212121212121212121212121 5 5 5 5 5 8 8 8 8 8 8 8 8
```