难度：1星：省三必会。2星：省二尽可能会，省一必会。3星：省一尽可能会。4星：国 2 以上要会

题目：

答案二分：https://hydro.ac/d/shallowdream/p/33 模版

求阶乘：https://www.lanqiao.cn/problems/2145/learning/ 2星

数列分段：https://www.luogu.com.cn/problem/P1182 2星

深秋的苹果：https://www.lanqiao.cn/problems/6279/learning 3星

成绩统计：https://www.lanqiao.cn/problems/19705/learning/ 3星

二分答案

暴力做法，枚举所有数字是否能达到某个值。

```cpp
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+10;
int a[N];
int main(){
    ll n,k;
    cin>>n>>k;
    for(int i=1;i<=n;i++){
        cin>>a[i];
    }
    for(int i=1;i<=k+N;i++){
        ll x=k;
        for(int j=1;j<=n;j++){
            if(a[j]<i){
                x=x-(i-a[j]);
            }
            if(x<0){
                cout<<i-1;
                return 0;
            }
        }
    }
}
```

```java
import java.util.Scanner;

public class Main {
    static final int N = 100010;
    static int[] a = new int[N];

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        long n, k;
        n = sc.nextLong();
        k = sc.nextLong();

        for (int i = 1; i <= n; i++) {
```

```
            a[i] = sc.nextInt();
        }

        for (int i = 1; i <= k + N; i++) {
            long x = k;
            for (int j = 1; j <= n; j++) {
                if (a[j] < i) {
                    x -= (i - a[j]);
                }
                if (x < 0) {
                    System.out.println(i - 1);
                    return;
                }
            }
        }
    }
}

```

```
N = 10**5 + 10
a = [0] * N

def main():
    n, k = map(int, input().split())
    a[1:n+1] = map(int, input().split())

    for i in range(1, k + N + 1):
        x = k
        for j in range(1, n + 1):
            if a[j] < i:
                x -= (i - a[j])
            if x < 0:
                print(i - 1)
                return

if __name__ == "__main__":
    main()

```

通过我们的暴力代码，可以发现，随着 i 变大，每次 j 循环中，消耗的 x 是变多的。

因此便具有一个单调性函数，即可拓展为二分，便是单调性（二段性），舍去不需要的一半。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+10;
int a[N];
ll n,k;
bool check(ll mid){
    ll x=k;
    for(int i=1;i<=n;i++){
        if(a[i]<mid){
            x=x-(mid-a[i]);
        }
        if(x<0){
            return false;
```

```
        }
    }
    return true;
}
int main(){
    cin>>n>>k;
    for(int i=1;i<=n;i++){
        cin>>a[i];
    }
    ll l=1,r=(ll)1e14;
    while(l<r){
        ll mid=l+r+1>>1;
        if(check(mid)) l=mid;
        else r=mid-1;
    }
    cout<<l;
}
```

数列分段

暴力枚举每一段的最小值，每一段的和你是不能超过这个值的。看看能分成多少段。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+10;
ll a[N];
ll n,m;
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++) cin>>a[i];
    ll ans=0;
    for(int i=1;i<=10000;i++){
        int cnt=1;
        ll sum=0;
        for(int j=1;j<=n;j++){
            if(a[j]>i){
                cnt=-1;
                break;
            }
            if(sum+a[j]<=i){
                sum=sum+a[j];
            }else{
                sum=a[j];
                cnt++;
            }
        }
        if(cnt!=-1){
            if(cnt<=m){
                cout<<i;
                break;
            }
        }
    }
}
```

```java
import java.util.Scanner;

public class Main {
    static final int N = 100010;
    static long[] a = new long[N];
    static long n, m;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextLong();
        m = sc.nextLong();

        for (int i = 1; i <= n; i++) {
            a[i] = sc.nextLong();
        }

        for (int i = 1; i <= 10000; i++) {
            int cnt = 1;
            long sum = 0;
            boolean valid = true;

            for (int j = 1; j <= n; j++) {
                if (a[j] > i) {
                    valid = false;
                    break;
                }
                if (sum + a[j] <= i) {
                    sum += a[j];
                } else {
                    sum = a[j];
                    cnt++;
                }
            }

            if (valid && cnt <= m) {
                System.out.println(i);
                break;
            }
        }
    }
}

```

```python
def main():
    n, m = map(int, input().split())
    a = [0] + list(map(int, input().split()))  # 1-based index

    for i in range(1, 10001):
        cnt = 1
        total = 0
        valid = True

        for j in range(1, n + 1):
            if a[j] > i:
                valid = False
                break
            if total + a[j] <= i:
```

```
                total += a[j]
            else:
                total = a[j]
                cnt += 1

        if valid and cnt <= m:
            print(i)
            break

if __name__ == "__main__":
    main()

```

二分

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+10;
ll a[N];
ll n,m;
bool check(ll mid){
    int cnt=1;
    ll sum=0;
    for(int j=1;j<=n;j++){
        if(a[j]>mid){
            return false;
        }
        if(sum+a[j]<=mid){
            sum=sum+a[j];
        }else{
            sum=a[j];
            cnt++;
        }
    }
    return cnt<=m;
}
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++) cin>>a[i];
    ll ans=0;
    ll l=1,r=(ll)(1e14);
    while(l<r){
        ll mid=l+r>>1;
        if(check(mid)) r=mid;
        else l=mid+1;
    }
    cout<<l;
}
```

```
import java.util.Scanner;

public class Main {
    static final int N = 100010;
    static long[] a = new long[N];
    static long n, m;

```

```
    static boolean check(long mid) {
        int cnt = 1;
        long sum = 0;

        for (int j = 1; j <= n; j++) {
            if (a[j] > mid) {
                return false;
            }
            if (sum + a[j] <= mid) {
                sum += a[j];
            } else {
                sum = a[j];
                cnt++;
            }
        }
        return cnt <= m;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextLong();
        m = sc.nextLong();

        for (int i = 1; i <= n; i++) {
            a[i] = sc.nextLong();
        }

        long l = 1, r = (long) 1e14;
        while (l < r) {
            long mid = (l + r) / 2;
            if (check(mid)) {
                r = mid;
            } else {
                l = mid + 1;
            }
        }

        System.out.println(l);
    }
}

```

```
def check(mid):
    cnt = 1
    total = 0

    for j in range(1, n + 1):
        if a[j] > mid:
            return False
        if total + a[j] <= mid:
            total += a[j]
        else:
            total = a[j]
            cnt += 1

    return cnt <= m
```

```python

if __name__ == "__main__":
    n, m = map(int, input().split())
    a = [0] + list(map(int, input().split()))  # 1-based index

    l, r = 1, int(1e14)
    while l < r:
        mid = (l + r) // 2
        if check(mid):
            r = mid
        else:
            l = mid + 1

    print(l)

```