

1

```
void reverseArray(int array[] , int size ){
    int l=0,r=size-1;
    while(l<r){
        int tmp=array[l];
        array[l++]=array[r];
        array[r--]=tmp;
    }
}
```

2

```
int m91(int n){
    printf("%d ",n);
    if(n>100){
        return n-10;
    }
    else{
        return m91(m91(n+11));
    }
}
```

3

```
void check( int matrix[][LEN] , int sum_row[] , int sum_col[] , int rows ,
int cols ){
    int err_i=-1,err_j=-1;
    for(int i=0;i<rows;++i){
        int sum=0;
        for(int j=0;j<cols;++j){
            sum+=matrix[i][j];
        }
        if(sum!=sum_row[i]){
            err_i=i;
        }
    }
    for(int i=0;i<rows;++i){
        int sum=0;
        for(int j=0;j<cols;++j){
            sum+=matrix[j][i];
        }
        if(sum!=sum_col[i]){
            err_j=i;
        }
    }
}
```

```
    if(err_i!=-1&&err_j!=-1)
        printf("%d %d %d",err_i,err_j,matrix[err_i][err_j]);
    else{
        printf("right");
    }
}
```

4

```
void recombination(char str1[], char str2[]){
    char str3[128] = {0};
    int idx1, idx2, idx3;
    idx1 = idx2 = idx3 = 0;
    while(str2[idx2] != '\0'){
        int cnt = str2[idx2] - '0';
        while(cnt--){
            str3[idx3++] = str1[idx1++];
        }
        str3[idx3++] = str2[idx2];
        idx2++;
    }
    while(str1[idx1] != '\0'){
        str3[idx3++] = str1[idx1++];
    }
    strcpy(str1, str3);
}
```

5

```
int check(int k, int n, int m, int* a){
    int sum=0;
    for(int i=0; i<n; i++){
        sum += a[i] / k;
    }
    if(sum >= m) return 1;
    else return 0;
}

int main(){
    int n, m;
    scanf("%d %d", &n, &m);
    int a[n];
    for(int i=0; i<n; i++){
        scanf("%d", &a[i]);
    }
    int max = 2000000;
    int min = 0;
    while(max >= min){
```

```
    int k = min + (max-min) / 2;
    if(k==0||check(k, n, m, a)){
        min = k + 1;
    }else{
        max = k - 1;
    }
}
printf("%d\n", max);
return 0;
}
```