**NỘI DUNG ĐÍNH CHÍNH SÁCH**

**TUYỂN TẬP ĐỀ THI HSG CẤP TỈNH THCS**

## 1. Bài 2. Đếm ước (đề Phú Thọ) – phần lời giải

|  |  |  |
| --- | --- | --- |
| Trang | Nội dung hiện tại trong sách | Sửa lại |
| 141 | for (int x=1; x <= n; x++)  if (cnt[x] == mmax) | for (int x=1; x < n; x++)  if (cnt[x] == mmax) |
| 142 | for i in range(1,n+1):  if (cnt[x] == mmax) | for i in range(1,n):  if (cnt[x] == mmax) |

## 2. Bài 5. Khảo sát TBZ (đề Nam Định) – phần lời giải

|  |  |  |
| --- | --- | --- |
| Trang | Nội dung hiện tại trong sách | Sửa lại |
| 110 | cin >> n;  for (int i=1; i<=n; i++)  {  ll x;  cin >> x;  a[i] = a[i-1] + x;  }  sort(a,a+n+1);  int d=1;  for (int i=1; i<=n; i++)  {  if (a[i] == a[i-1]) d++;  else  {  ans += d \* (d - 1) / 2;  d = 1;  }  }  cout << ans; | cin >> n;  for (int i=1; i<=n; i++)  {  ll x;  cin >> x;  a[i] = a[i-1] + x;  }  sort(a,a+n+1);  int d=1;  a[n+1] = a[n] + 1;  for (int i=1; i<=n+1; i++)  {  if (a[i] == a[i-1]) d++;  else  {  ans += d \* (d - 1) / 2;  d = 1;  }  }  cout << ans; |
| 110 | import sys  sys.stdin = open("tbz.inp","r")  sys.stdout = open("tbz.out","w")  n = int(input())  a = [0] + list(map(int,input().split()))  for i in range(1,n+1): a[i] = a[i-1] + a[i];  a.sort()  ans = 0;  d = 1;  for i in range(1,n+1):  if (a[i] == a[i-1]): d += 1;  else:  ans += d \* (d - 1) // 2;  d = 1;  print(ans) | import sys  sys.stdin = open("tbz.inp","r")  sys.stdout = open("tbz.out","w")  n = int(input())  a = [0] + list(map(int,input().split()))  for i in range(1,n+1): a[i] = a[i-1] + a[i];  a.sort()  a = a + [a[n]+1]  ans = 0;  d = 1;  for i in range(1,n+2):  if (a[i] == a[i-1]): d += 1;  else:  ans += d \* (d - 1) // 2;  d = 1;  print(ans) |

## 3. Bài 3. Thừa số nguyên tố (đề Thanh Hóa) – phần lời giải

|  |  |  |
| --- | --- | --- |
| Trang | Nội dung hiện tại trong sách | Sửa lại |
| 196 | #include <bits/stdc++.h>  #define ll long long  using namespace std;  ll n;  void phantich(ll n)  {  for (ll i = 2; i \* i <= n; i++)  if (n % i == 0)  {  int d = 0;  while (n % i == 0)  {  d++;  n /= i;  }  cout << i << " " << d << '\n';  }  if (n > 1) cout << n << " " << 1;  }  int main()  {  freopen("thuasont.inp","r",stdin); freopen("thuasont.out","w",stdout);  cin >> n;  phantich(n);  return 0;  } | #include <bits/stdc++.h>  #define ll long long  #define N 1000000  using namespace std;  ll n;  int p[N+2];  int dem = 0;  void phantich(ll n)  {  for (ll i = 2; i \* i <= n; i++)  if (n % i == 0)  {  dem++;  int d = 0;  while (n % i == 0)  {  d++;  n /= i;  p[i]++;  }  }  if (n > 1) dem++;  cout << dem << '\n';  for (int i=2; i<=N; i++)  if (p[i] > 0) cout << i << " " << p[i] << "\n";  if (n > 1) cout << n << " " << 1;  }  int main()  {  freopen("thuasont.inp","r",stdin);  freopen("thuasont.out","w",stdout);  cin >> n;  phantich(n);  return 0;  } |
| 197 | import sys  sys.stdin = open("thuasont.inp","r")  sys.stdout = open("thuasont.out","w")  n = int(input())  def phantich(n):  k = int(n \*\* 0.5)  for i in range(2,k+1):  if (n % i == 0):  d = 0;  while (n % i == 0):  n //= i  d += 1;  print(i, d)  if (n > 1): print(n,1)  phantich(n) | import sys  sys.stdin = open("thuasont.inp","r")  sys.stdout = open("thuasont.out","w")  n = int(input())  p = [0] \* 1000002  dem = 0  def phantich(n):  global dem  k = int(n \*\* 0.5)  for i in range(2,k+1):  if (n % i == 0):  dem += 1;  d = 0;  while (n % i == 0):  n //= i  d += 1;  p[i] += 1;  if (n > 1): dem += 1;  print(dem)  for i in range(2,1000001):  if (p[i] > 0): print(i,p[i])  if (n > 1): print(n,1)  phantich(n) |

## 4. Bài 3. Bánh mỳ và bánh rán (Đề HSG Quảng Ninh) – phần lời giải

*Toàn bộ phần hướng dẫn giải trong sách được thay thế bởi những nội dung dưới đây:*

**\* Thuật toán đề xuất:**

Trước hết hãy quan sát một đoạn con thuộc dãy như hình vẽ:

A black and white text

Description automatically generated

Từ hình vẽ có thể thấy rằng, số lượng số 1 sau khi đổi k phần tử liên tiếp thành 1 sẽ là .

Để có thể tính nhanh được và có thể sử dụng hai mảng , với ý nghĩa sau:

* = độ dài đoạn con chứa số 1 liên tiếp kết thúc tại vị trí tính từ trái qua phải.
* = độ dài đoạn con chứa số 1 liên tiếp kết thúc tại vị trí tính từ phải qua trái.

Như vậy, chỉ cần duyệt phần tử liên tiếp từ vị trí đến vị trí (với i = k 🡪 n) thì đoạn con liên tiếp bằng giá trị 1 sẽ là: . So sánh tổng này với kết quả để cập nhật lớn nhất.

**Độ phức tạp:**

**Chủ đề: *Duyệt***

**\* Chương trình minh họa bằng ngôn ngữ lập trình C++:**

#include <bits/stdc++.h>

#define N 1000000

using namespace std;

int a[N+2],dpL[N+2],dpR[N+2],F[N+2];

int n, k, ans=0;

int main()

{

freopen("donu.inp","r",stdin);

freopen("donu.out","w",stdout);

cin >> n >> k;

for (int i=1; i<=n; i++) cin >> a[i];

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

if (a[i] == 1) dpL[i] = dpL[i-1] + 1;

for (int i=n; i >= 0; i--)

if (a[i] == 1) dpR[i] = dpR[i+1] + 1;

int d = 0;

for (int i=k; i<=n; i++)

ans = max(ans,dpL[i-k] + k + dpR[i+1]);

cout << ans;

return 0;

}

**\* Chương trình minh họa bằng ngôn ngữ lập trình Python:**

import sys

sys.stdin = open("donu.inp","r")

sys.stdout = open("donu.out","w")

n, k = map(int,input().split())

a = [0] + list(map(int,input().split()))

dpL = [0] \* (n + 2)

dpR = [0] \* (n + 2)

for i in range(1, n+1):

if (a[i] == 1): dpL[i] = dpL[i-1] + 1;

for i in range(n, 0, -1):

if (a[i] == 1): dpR[i] = dpR[i+1] + 1;

d = 0;

ans = 0;

for i in range(k, n + 1):

ans = max(ans,dpL[i-k] + k + dpR[i+1]);

print(ans)

Bộ test đã được cập nhật lại.