

# beCP 2022

## Taak 1.2: Rangemin (rangemin)

---

Auteur: Damien Galant

Vorbereiding: Damien Galant, Victor Dheur, Pierre Vandenhove

Maximale uitvoeringsduur: 1 s Geheugenlimiet: 512 MB

---

*Opmerking: Dit is een interactieve taak. Kijk de speciale instructies om je programma te implementeren, compileren en testen na. Aarzel niet om een begeleider om hulp te vragen.*

Er zijn  $n$  verschillende positieve getallen  $a_0, a_1, \dots, a_{n-1}$ . Jouw doel is om de *index* van het kleinste van die getallen te vinden. Je hebt echter geen directe toegang tot deze getallen, maar enkel tot de functie `range_min(i)` (zie later). Je wil deze functie zo *weinig mogelijk* oproepen.

### Op te roepen functies

De functie `range_min(i)` geeft de minimale *waarde* terug van de getallen  $a_i, a_{i+1}, \dots, a_{i+k-1}$ . Eens je de *index* van het minimum van  $a_0, a_1, \dots, a_{n-1}$  gevonden hebt moet je de functie `found_min(i)` eenmalig oproepen met de *index* van het minimum.

### Te implementeren functies

Je moet de functie `find_min(n,k)` implementeren. Deze functie krijgt de waardes van  $n$  en  $k$ . Je moet de functie `range_min` oproepen om de *index* van het minimum van  $a_0, a_1, \dots, a_{n-1}$  te vinden. Eens je dat gevonden hebt, moet je `found_min` eenmalig oproepen met deze *index*, en dan je programma beëindigen.

*Opmerking: als je `range_min` niet oproept voor een input zal je een “wrong answer” verdict krijgen, zelfs als je (puur toevallig) de correcte index van het minimum vindt.*

### Algemene limieten

Alle waardes  $a_0, a_1, \dots, a_{n-1}$  zijn *verschillende* gehele getallen zodat  $1 \leq a_i \leq 1\,000\,000$  for alle  $0 \leq i < n$ .

Je can de `range_min` functie maximaal 100 000 keer oproepen (dit is een technische beperking, een oplossing die `range_min` zo veel keer oproept zou een heel lage score hebben, zie verder bij “Score”).

## Bijkomende beperkingen

Subtaak	Punten	Beperkingen
A	20	$n = 100, k = 1$
B	20	$n = 100, k = 2$
C	60	$n = 1000, 1 \leq k \leq 500$

## Voorbeeld 1

Dit is een mogelijke volgorde van events met de veronderstelling dat  $n = 5$ ,  $k = 2$ ,  $a_0 = 8$ ,  $a_1 = 10$ ,  $a_2 = 4$ ,  $a_3 = 5$  and  $a_4 = 9$ .

Jouw oproep	Antwoord
<code>range_min(0)</code>	$\min(8, 10) = 8$
<code>range_min(3)</code>	$\min(5, 9) = 5$
<code>range_min(2)</code>	$\min(4, 5) = 4$
<code>found_min(2)</code>	Index gevonden!

*Opmerking: let er op dat dit klein voorbeeld geen deel uitmaakt van eender welke subtaak (A, B or C).*

## Score

De score voor deze taak wordt als volgt bepaald:

- De index van het minimum moet voor elke test van een subtask gevonden worden om punten te krijgen voor die subtask.
- Als dit het geval is, dan schrijven we  $S$  voor het aantal oproepen naar `range_min` dat jouw oplossing gemaakt heeft, en  $B$  voor het aantal oproepen naar `range_min` dat de oplossing van de beCP jury gemaakt heeft. De *verhouding* voor deze test is  $\min(B/S, 1)$ . Op CMS wordt deze verhouding voor elke testcases weergegeven waar je de correct index van het minimum gevonden hebt.
- Tot slot is je score voor een subtask gegeven door het totaal aantal punten beschikbaar voor deze subtask, vermenigvuldigd met  $0.2 + 0.8 * r$ , waar  $r$  de *laagste* verhouding tussen de tests voor deze subtask is.

Bijvoorbeeld, als de laagste verhouding voor de tests van subtask B gelijk is aan 0.4, dan is je score voor subtask B gelijk aan  $20 + (0.2 + 0.8 \times 0.4) = 10.4$ .

## Compilatie

Dit is een interactieve taak. Volg de volgende stappen om je programma te compileren en manueel te testen. Indien nodig, vraag voor hulp van een verantwoordelijke.

- Open een terminal;
- Om de lijst van bestanden in een folder weer te geven, gebruik het commando `ls`;
- Om van folder te veranderen naar deze taak, gebruik het commando `cd folder_name`;
- Om je C++ programma te compileren en uit te voeren, gebruik volgende commando's:
  - Compileren  
`g++ -std=c++11 -Wall -Werror -Wshadow grader.cpp rangemin.cpp`
  - Uitvoeren: `./a.out < input.in`

Het resultaat van je programma wordt naar de console geprint: „FOUND THE INDEX AFTER X CALLS” als de index van het minimum gevonden werd en „WRONG INDEX WAS GIVEN” als de index die je gaf fout is. Andere berichten kunnen geprint worden als je argumenten naar `range.min` of `found_min` fout zijn. Als je het bestand `input.in` wil aanpassen, weet dan dat de eerste lijn  $n$  en  $k$  bevat, en de tweede lijn  $n$  getallen  $a_0, a_1, \dots, a_{n-1}$ . Zorg er voor dat deze getallen verschillend zijn, en tussen 1 en 1 000 000 liggen!

### Opmerkingen

- Je dient enkel het bestand `rangemin.cpp` in te dienen. Het moet de functie `find_min` implementeren, zoals eerder beschreven.
- **Print niets naar de console in je programma (`printf`, `cout`).** Je kan `cerr` in C++ gebruiken om debug informatie weer te geven.