

Started on	Sunday, 30 March 2025, 9:56 PM
State	Finished
Completed on	Sunday, 30 March 2025, 11:36 PM
Time taken	1 hour 40 mins
Marks	300.00/440.00
Grade	7.50 out of 11.00 (68%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama file : **palindromlagi.c**

Setelah melewati **tiga kali pelatihan rahasia**, para Nimons akhirnya mulai memahami peraturan permainan di dunia mereka. Pada awalnya, konsep bilangan **palindrom** terasa asing bagi mereka, tetapi dengan bimbingan **Gro**, seorang mentor bijak, mereka perlahan mulai menguasainya.

Di pelatihan pertama, Gro mengajarkan tentang **bilangan yang tetap sama jika dibaca dari depan maupun belakang**. Para Nimons dengan cepat mengenali contoh sederhana seperti **121, 333, dan 898**. Namun, ketika dihadapkan pada bilangan besar, beberapa Nimons masih kebingungan.

Pelatihan terakhir ini adalah yang paling menantang. Gro memberikan mereka sejumlah kata dan meminta mereka untuk **menentukan apakah kata tersebut dapat dibuat menjadi palindrom** dengan menggeser huruf-hurufnya. Para Nimons harus berpikir cepat dan menggunakan semua yang telah mereka pelajari untuk menyelesaikan tantangan ini.

Bantulah para Nimons dalam ujian terakhir ini ! Buatlah program yang menerima input sebuah kalimat, dan mengeluarkan **"YES"** atau **"NO"** tergantung dengan apakah mereka palindrom!

Note :

Gunakan **string.h** untuk membaca input kata.

Gunakan char[1001] sebagai placeholder dari variable kalian, dan gunakan"%s" untuk membacanya.

Example:

char s[1001];

scanf("%s", s);

Contoh Masukan dan keluaran

No	Masukan	Keluaran	Keterangan
1	aba	YES	Memang palindrom
2	abc	NO	Bukan palindrom
3	abb	YES	Bisa diubah menjadi "bab" yang adalah palindrom



[palindromlagi.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.57 MB
3	10	Accepted	0.00 sec, 1.58 MB
4	10	Accepted	0.00 sec, 1.60 MB
5	10	Accepted	0.00 sec, 1.65 MB
6	10	Accepted	0.00 sec, 1.57 MB
7	10	Accepted	0.00 sec, 1.57 MB
8	10	Accepted	0.00 sec, 1.50 MB
9	10	Accepted	0.00 sec, 1.57 MB
10	10	Accepted	0.00 sec, 1.65 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: kodekeamanan.c

Di sebuah laboratorium rahasia milik **Gro**, para Nimons sedang menghadapi tantangan besar! Mereka harus membuka **brankas super rahasia** yang hanya bisa dibuka dengan **kode keamanan khusus**.

Kebin, sang pemimpin Nimons, menemukan petunjuk bahwa kode ini bisa didapatkan dari **dua angka dalam sebuah daftar** yang jika dijumlahkan akan menghasilkan target tertentu. Namun, mereka tidak boleh menggunakan **angka yang sama dua kali!**

Stewart, Pop, dan Toto mencoba semua kombinasi angka, tetapi hasilnya selalu berantakan! **Dr. Neroifa**, ilmuwan jenius, akhirnya meminta bantuan Anda untuk menulis program yang bisa menemukan **dua angka** dalam daftar yang jika dijumlahkan akan menghasilkan **kode rahasia**.

Format Masukan:

- Sebuah bilangan bulat **N** ($2 \leq N \leq 10^4$).
- Sejumlah **N** angka yang merupakan elemen array nums ($-10^9 \leq \text{angka} \leq 10^9$)..
- Sebuah bilangan bulat **target** yang merupakan hasil penjumlahan dua angka yang dicari ($-10^9 \leq \text{target} \leq 10^9$).
- Jika ada lebih dari satu solusi **pilih angka paling minimal**.
- Anda **tidak boleh menggunakan elemen yang sama dua kali**.

Format Keluaran:

- Dua indeks dari nums yang memenuhi kriteria, dalam **urutan menaik**.

Contoh Masukan dan Keluaran:

No	Masukan	Keluaran	Keterangan
1.	4 2 7 11 15 9	[0, 1]	nums[0] + nums[1] = 2 + 7 = 9, maka kita mengembalikan [0,1].
2.	2 3 3 6	[0, 1]	nums[0] + nums[1] = 3 + 3 = 6, maka kita mengembalikan [0, 1].
3.	3 3 2 4 6	[1, 2]	nums[1] + nums[2] = 2 + 4 = 6, maka kita mengembalikan [1, 2].

Notes:

- Pastikan setiap akhir baris tidak mengandung **spasi** (" ")!
- Pastikan setiap output diakhiri oleh **endline** ("\\n")!

C

⌵

 [kodekeamanan.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.64 MB
2	10	Accepted	0.00 sec, 1.60 MB
3	10	Accepted	0.00 sec, 1.71 MB
4	10	Accepted	0.00 sec, 1.66 MB

Time limit	1 s
Memory limit	64 MB

Karena Asisten sangat baik hatinya, asisten menyarankan Gro untuk tidak mempersulit Nimons dalam mengerjakan task ini. Oleh karena itu, Gro akan menganggap task ini sebagai task bonus yang tidak wajib dikerjakan.



Nama File: VirusGro.c

Tuan Gro adalah seorang penjahat super. Ia bersama dengan rekannya, Dr. Neroifa, sedang mengembangkan sebuah senjata biologis berupa sebuah virus mematikan. Untuk mengetahui kemampuan senjata ini, mereka melakukan uji coba terhadap sebuah deretan rumah sepanjang n . Rumah ke- i ($1 \leq i \leq n$), memiliki nilai afinitas terhadap virus sebesar a_i ($-10^9 \leq a_i \leq 10^9$).

Tuan Gro hanya memiliki k buah sampel virus untuk disebarakan sehingga ia dapat memilih d ($1 \leq d \leq k$) buah rumah sebagai titik penyebaran awal. Tuan Gro juga memiliki kemampuan untuk mengontrol penyebaran virus ini sehingga selama masih ada rumah belum terinfeksi, ia harus memilih rumah belum terinfeksi manapun yang bertetangga dengan rumah yang sudah terinfeksi untuk diinfeksi oleh virus.

Nilai keefektifan dari penyebaran virus ini didefinisikan sebagai jumlah dari nilai afinitas d buah rumah yang dipilih di awal dan nilai afinitas dari rumah terakhir yang terkena penyebaran virus. Tuan Gro meminta bantuanmu untuk menghitung **nilai keefektifan maksimal** yang dapat diperoleh dengan pemilihan yang optimal.

Format Masukan:

- Baris pertama berisi bilangan bulat positif n dan k ($2 \leq n \leq 5000$; $1 \leq k < n$) yang mewakili banyak rumah dan banyak sampel virus.
- Baris kedua berisi n buah bilangan bulat a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) yang mewakili nilai afinitas dari rumah ke- i .

Format Keluaran:

- Cetak nilai keefektifan maksimal yang mungkin didapatkan.

Contoh Masukan dan Keluaran:

No	Masukan	Keluaran	Keterangan
1.	3 1 1 2 3	5	Untuk $k = 1$, nilai keefektifan maksimum diperoleh dengan memilih rumah ke-2 sebagai titik penyebaran awal, kemudian berturut-turut memilih rumah ke-1 dan rumah ke-3 untuk diinfeksi. Jadi, nilai keefektifan adalah $a_2 + a_3 = 5$
2.	5 3 1 2 -3 -4 5	8	Untuk $k = 3$, nilai keefektifan maksimum dapat diperoleh dengan memilih 2 buah rumah saja sebagai titik penyebaran awal, yaitu rumah ke-1 dan ke-5. Selanjutnya, lakukan infeksi terhadap rumah ke-4, ke-3, dan ke-2 secara berturut-turut sehingga nilai keefektifannya adalah: $a_1 + a_2 + a_5 = 8$

Keterangan:

- Pastikan setiap output diakhiri dengan **newline ("**\n**")** untuk memastikan format keluaran yang benar.
- Gunakan tipe data **long long** untuk melakukan perhitungan agar menghindari *overflow*.

Question **4**

Incorrect

Mark 0.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama file: **BinarySearch.c**

Gro memiliki daftar angka yang sudah diurutkan. Ia ingin mencari apakah sebuah angka tertentu ada dalam daftar tersebut menggunakan **Binary Search**. Bantu Gro membuat program untuk menemukan **indeks angka** dalam daftar atau mengembalikan **-1** jika tidak ditemukan.

Input:

- 1. **N** – Sebuah bilangan bulat ($1 \leq N \leq 10^5$) yang menyatakan jumlah angka dalam daftar.
- 2. **List Angka** – N bilangan bulat yang sudah **terurut naik (ascending order)**.
- 3. **Q** – Sebuah bilangan bulat ($1 \leq Q \leq 10^4$) yang menyatakan jumlah pencarian angka yang akan dilakukan.
- 4. **Q Query** – Tiap query berupa satu bilangan bulat yang ingin dicari dalam daftar.

Output:

- Untuk setiap query, jika angka ditemukan dalam daftar, cetak **indeksnya (0-based index)**.
- Jika angka **tidak ditemukan**, cetak **-1**.

Contoh input/output:

No	Masukan	Keluaran	Keterangan
1.	10 1 2 2 3 3 3 4 4 5 6 3 3 4 7	[3, 6, -1]	Angka 3 ditemukan pertama kali pada array pada indeks 3. Sama halnya, dengan angka 4, ditemukan pada indeks 6 pertama kali. Angka 7 tidak ditemukan.
2.	1 42 2 42 7	[0, -1]	Angka 42 ditemukan pada array [42] sehingga indeks 0 Angka 7 tidak ditemukan sehingga indeks -1

Pastikan setiap output diakhiri oleh **endline ("**\n**")!**

Catatan:

- **Jika angka ditemukan lebih dari sekali (duplikat), kembalikan indeks terendah.**
- **Array sudah dalam keadaan terurut, jadi tidak perlu melakukan pengurutan ulang.**

Hint: Buat fungsi `BinarySearch(int arr[], int left, int right, int target)`

C ▾

Score: 0

Blackbox

Score: 0

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.00 sec, 1.53 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Suatu hari di laboratorium Gro, para Nimons sedang menyusun pisang-pisang kesayangan mereka dalam sebuah baris. Kebin, Stewart, dan Pop ditugaskan untuk mengatur ulang pisang-pisang tersebut untuk eksperimen terbaru Gru.

Gro masuk ke laboratorium dengan membawa sebuah alat aneh. "Nimons! Hari ini kita akan melakukan eksperimen 'Banana Rotator 3000'! Alat ini akan merotasi posisi pisang sebanyak k langkah ke kanan!"

Para Nimons terlihat kebingungan. Bantulah para Nimons untuk menyelesaikan eksperimen Gro! Buatlah program dengan nama file **kputaran.c** yang menerima input jumlah pisang N dan jumlah langkah rotasi k kemudian mengeluarkan output hasil putaran tersebut!

Contoh input/output:

No	Masukan	Keluaran	Keterangan
1	5 3 1 2 3 4 5	3 4 5 1 2	Jumlah elemen = 5 Jumlah geseran ke kanan = 3 Geseran pertama: 5 1 2 3 4 Geseran kedua: 4 5 1 2 3 Geseran ketiga: 3 4 5 1 2
2	5 0 1 2 3 4 5	1 2 3 4 5	Karena tidak ada geseran, maka hasil tetap
3	5 5 1 2 3 4 5	1 2 3 4 5	Karena jumlah geseran = panjang larik, maka hasilnya akan sama seperti larik awal.

Catatan:

- Pastikan setiap output diakhiri oleh endline (" \backslash n")!

Batasan:

- $1 \leq N \leq 10^4$
- $1 \leq k \leq 10^4$

C

▾

 [kputaran.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.50 MB
2	10	Accepted	0.00 sec, 1.55 MB
3	10	Accepted	0.00 sec, 1.56 MB
4	10	Accepted	0.00 sec, 1.63 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.62 MB
7	10	Accepted	0.00 sec, 1.50 MB
8	10	Accepted	0.00 sec, 1.49 MB
9	10	Accepted	0.00 sec, 1.49 MB
10	10	Accepted	0.00 sec, 1.60 MB