

---

**Started on** Thursday, 28 October 2021, 7:13 PM

**State** Finished

---

**Completed on** Friday, 12 November 2021, 9:50 PM

**Time taken** 15 days 2 hours

---

**Marks** 130.00/130.00

---

**Grade** **100.00** out of 100.00

Question **1**

Correct

Mark 70.00 out  
of 70.00

Time limit

1 s

Memory limit

64 MB

Buatlah program dalam file bernama **bigint.c** yang membaca sebuah 2 (dua) buah string yang diakhiri dengan tanda pagar ('#') yang masing-masing merepresentasikan sebuah integer besar (big integer) positif atau 0. Panjang setiap string (di luar tanda pagar) maksimum adalah 99, sehingga integer maksimum yang bisa direpresentasikan terdiri atas 99 digit. Panjang string minimum (di luar tanda pagar) adalah 1, yaitu jika integer yang direpresentasikan hanya terdiri atas sebuah digit. Selanjutnya program akan menampilkan hasil penjumlahan kedua integer tersebut.

Petunjuk:

- Gunakan stack untuk mensimulasikan penjumlahan tiap digit dari integer. Pakailah ADT Stack (file header [stack.h](#)) yang telah Anda kerjakan sebagai tugas pra-praktikum.
- Konversi masing-masing karakter dalam string menjadi sebuah integer dan dengan menggunakan stack of integer, push setiap integer ke dalam stack. Misalnya stack yang dihasilkan adalah S1 (untuk bilangan 1) dan S2 (untuk bilangan 2). Dengan demikian, bilangan yang merepresentasikan satuan (digit terakhir bilangan) akan berada di top dari stack.

Ingat: Karena string pada dasarnya adalah array of character, maka setiap elemen string dapat diakses seperti mengakses array of character (ingat: indeks terkecil dari 0).

- Setelah kedua bilangan masing-masing dipindahkan ke dalam sebuah stack, lakukan penjumlahan dan push hasil penjumlahan ke dalam suatu stack of integer yang lain, misalnya SPlus. Kali ini karena bilangan yang di-push dimulai dari satuan (digit terakhir), maka bilangan digit pertama akan berada di top stack. Ingat: Anda juga harus mengelola sisa dari hasil penjumlahan tiap digit. Contohnya: digit 9 + digit 8 akan menghasilkan digit 7 dan sisa penjumlahan 1 (yang ditambahkan pada penjumlahan digit berikutnya).
- Terakhir, cetak setiap digit SPlus mulai dari top stack ke bottom stack.

Contoh input/output:

Input	Output
12345678# 123456789#	135802467
0# 12345#	12345

C

 [bigint.c](#)

Score: 70

Blackbox

Score: 70

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.01 sec, 1.57 MB
3	10	Accepted	0.00 sec, 1.47 MB
4	10	Accepted	0.00 sec, 1.55 MB
5	10	Accepted	0.00 sec, 1.58 MB
6	10	Accepted	0.02 sec, 1.53 MB
7	10	Accepted	0.00 sec, 1.60 MB

Question **2**

Correct

Mark 60.00 out of 60.00

Time limit

1 s

Memory limit

64 MB

Dengan memanfaatkan ADT Stack yang telah Anda kerjakan sebagai tugas pra-praktikum (file header: [stack.h](#)) dan Token Machine yang dihasilkan pada praktikum sebelumnya, buatlah sebuah program dalam file bernama **ekspresi.c** yang membaca sebuah pita karakter yang hanya berisi bilangan positif dan 0 serta operator '+', '-', '\*', '/', dan '^' yang membentuk sebuah ekspresi aritmatika dalam notasi postfix dan mengevaluasi ekspresi yang terdapat pada pita karakter tersebut.

Secara lebih rinci, program memproses hasil dari mesin token sebagai berikut:

- Apabila menerima token bilangan, maka bilangan yang dibaca ditampilkan di layar dan dimasukkan ke dalam stack.
- Apabila menerima token operator, maka operan yang sesuai diambil dari stack, sub ekspresi yang harus dieksekusi dituliskan ke layar (format <operan1> <operator> <operan2>), hitung hasil sub ekspresi tersebut, tampilkan ke layar, dan masukkan ke dalam stack. Operan "/" diartikan sebagai pembagian bilangan bulat (div).
- Apabila pembacaan pita karakter telah selesai, hasil akhir ekspresi dituliskan ke layar.

Program akan menuliskan pesan "Ekspresi kosong" apabila pita karakter kosong.

Petunjuk:

- Untuk operasi '^' gunakan fungsi pow pada library math.h.
- Implementasi untuk [tokenmachine.h](#) tidak perlu dibuat ulang, manfaatkan fungsinya dari header yang tersedia pada praktikum yang lalu.
- Yang harus dikumpulkan: **ekspresi.c**

Contoh masukan dan keluaran program:

Input (Isi Pita Karakter)	Output (Ke Layar)
25 10 * 15 +.	25 10 25 * 10 250 15 250 + 15 265 Hasil=265
4 5 * 3 10 * 1 3 + / +.	4 5 4 * 5 20 3 10 3 * 10 30 1 3 1 + 3 4 30 / 4 7 20 + 7 27 Hasil=27
25 .	25 Hasil=25
.	Ekspresi kosong

C

 [ekspresi.c](#)

Score: 60

Blackbox

Score: 60

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.65 MB
3	10	Accepted	0.00 sec, 1.54 MB
4	10	Accepted	0.00 sec, 1.59 MB
5	10	Accepted	0.00 sec, 1.67 MB
6	10	Accepted	0.00 sec, 1.51 MB

[◀ Pra Praktikum 8](#)

Jump to...

[node.h ▶](#)