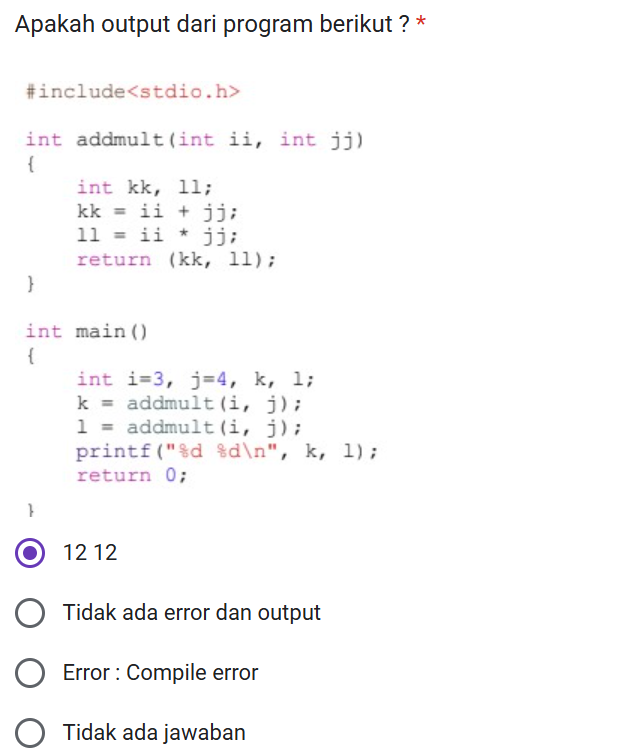
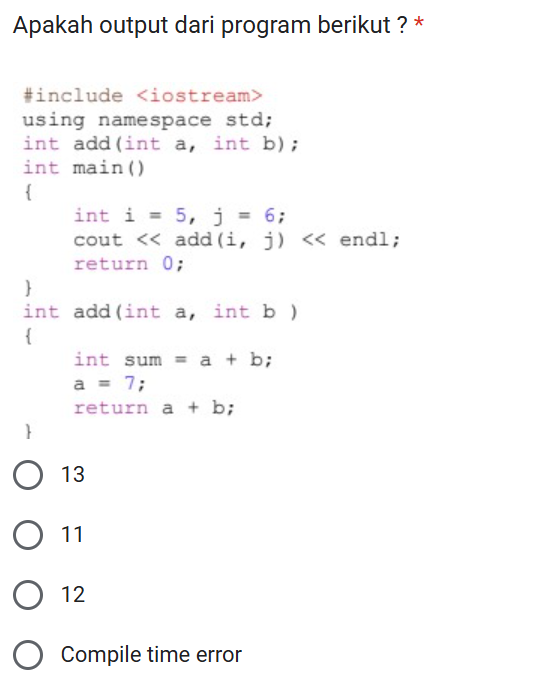
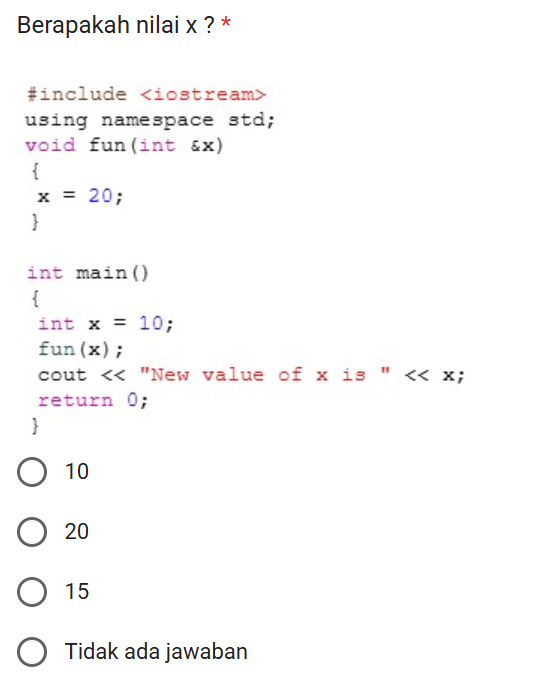
# **Penjelasan**



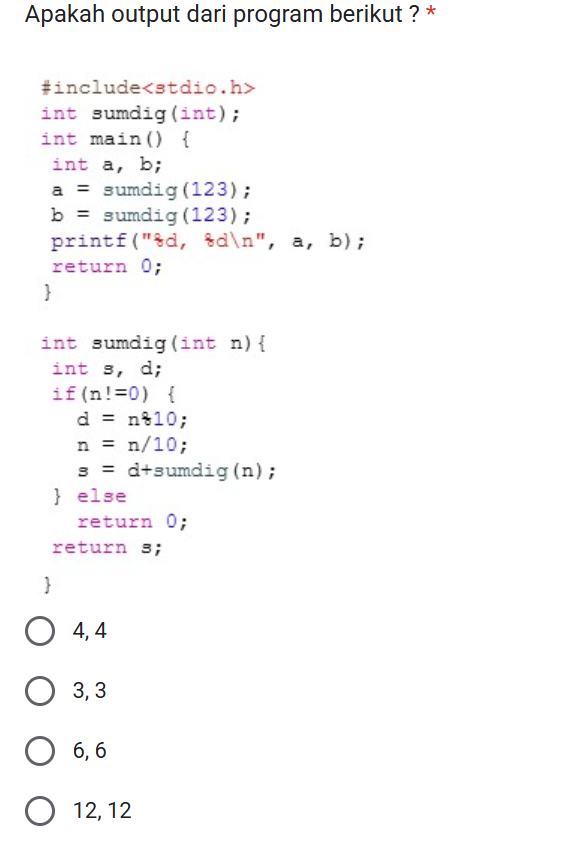
**Jawaban:** 12.12  
**Penjelasan:**



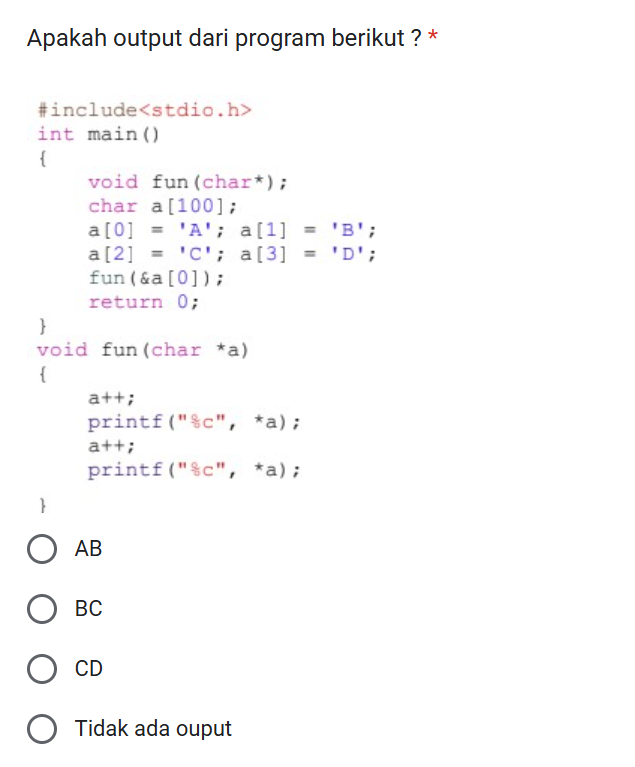
**Jawaban:** 1. 13  
**Penjelasan:** Fungsi add mengembalikan hasil penjumlahan a + b setelah a diubah menjadi 7. Karena a = 7 dan b = 6, hasilnya adalah 7 + 6 = 13. Perubahan nilai a tidak memengaruhi nilai i di main karena a adalah parameter lokal.



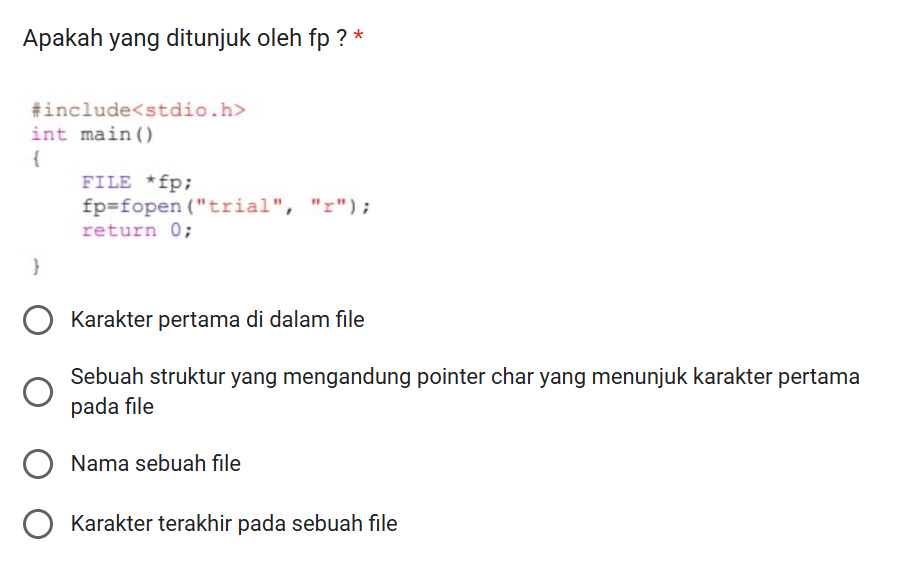
**Jawaban:** 1. 10  
**Penjelasan:** Variabel x di dalam fungsi main tidak terpengaruh oleh perubahan di fungsi fun karena x di fun adalah variabel lokal. Nilai x di main tetap 10, dan fungsi fun bahkan tidak dapat dijalankan karena sintaksnya salah (kurang tanda kurung penutup).



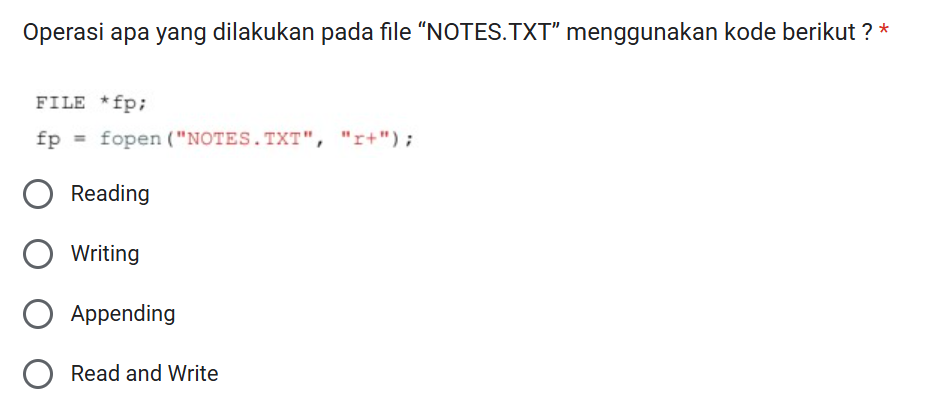
**Jawaban:** 3. 6,6  
**Penjelasan:** Fungsi sumdig menghitung jumlah digit dari bilangan n. Untuk n = 123, hasilnya adalah 1 + 2 + 3 = 6. Karena a dan b dipanggil dengan nilai yang sama (123), outputnya adalah 6, 6.



**Jawaban:** 2. BC  
**Penjelasan:** Fungsi fun menerima pointer ke elemen pertama array a. Setelah a++, pointer menunjuk ke elemen kedua ('B'), dan setelah a++ lagi, pointer menunjuk ke elemen ketiga ('C'). Jadi, outputnya adalah BC.

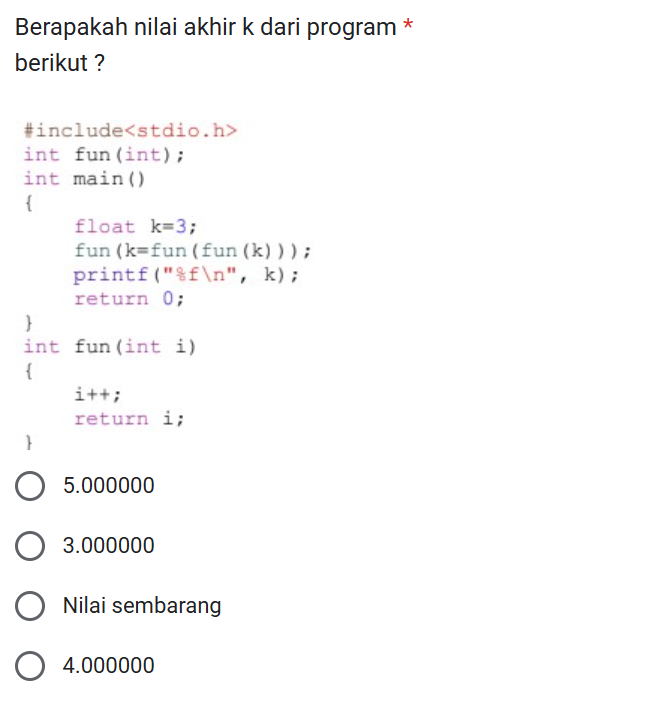


fp adalah pointer ke struktur FILE, yang menyimpan informasi tentang file yang dibuka, termasuk posisi pembacaan. Saat fopen("trial", "r") dijalankan, fp tidak langsung menunjuk ke karakter pertama, tetapi ke struktur yang mengelola akses file tersebut.



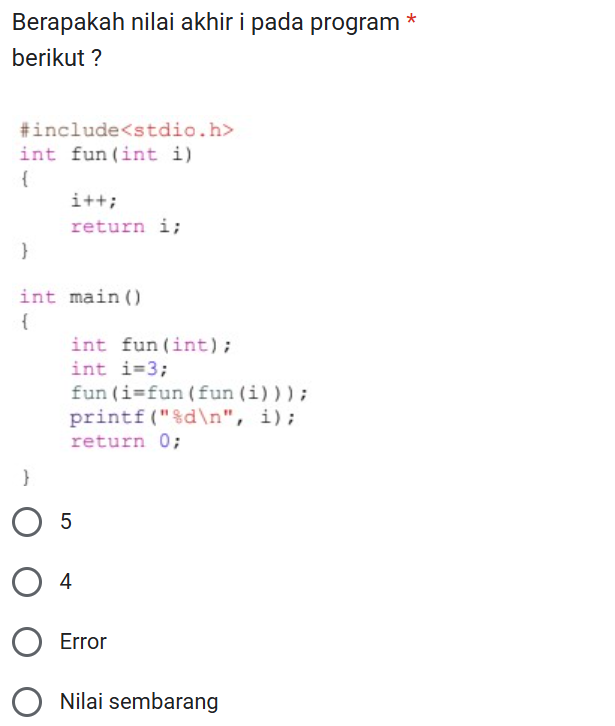
Jawaban yang benar : Read and Write

Karena r+ adalah operator untuk membaca dan menulis ke file.



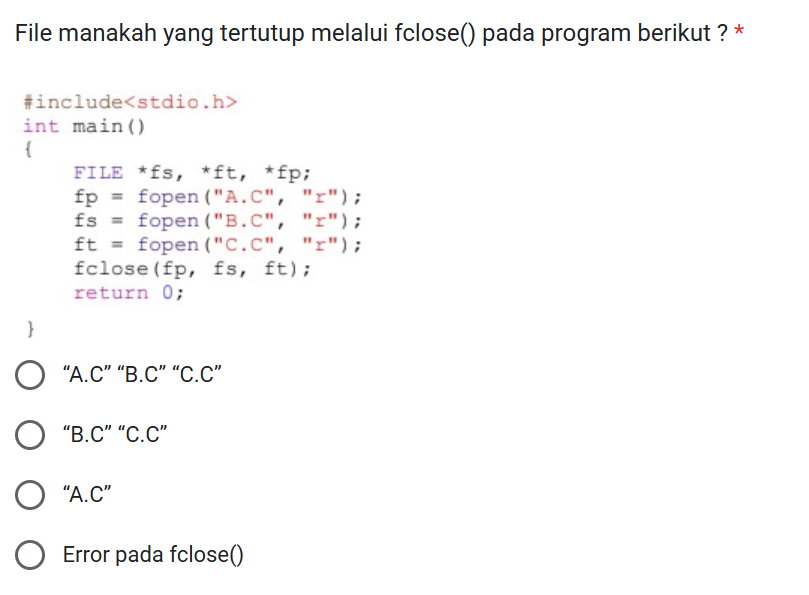
Jawaban yang benar adalah **5.000000**

Secara logika, k awalnya bernilai 3. Kemudian, fun(k) dipanggil, yang berarti fun(3), yang meningkatkan nilai i menjadi 4 dan mengembalikannya. Kemudian, fun(fun(k)) menjadi fun(4), yang meningkatkan i menjadi 5 dan mengembalikannya. Jadi, k akhirnya bernilai 5.0. Oleh karena itu, jawaban yang benar seharusnya adalah **"5.000000"**.



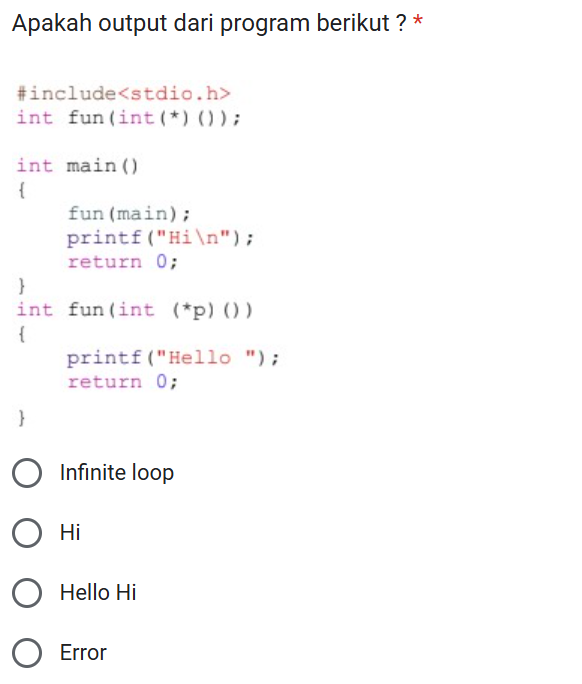
Jawaban yang benar adalah **5**.

Penjelasan: Nilai akhir i tetap **3**, bukan **4** seperti yang dipilih. Jawaban yang benar seharusnya adalah **3**, tetapi opsi ini tidak tersedia, yang mungkin menunjukkan kesalahan dalam opsi jawaban.



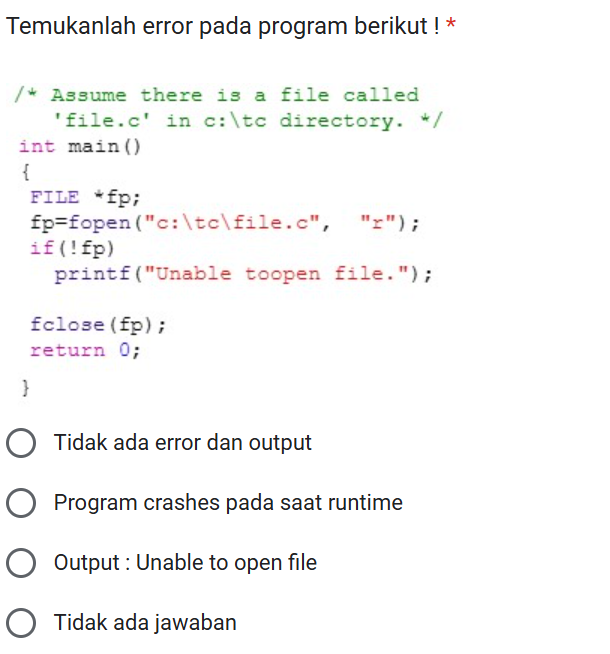
Jawaban yang benar adalah **"Error pada fclose()"**.

Fungsi fclose() hanya menerima satu argumen dalam satu pemanggilan, sedangkan pada kode di atas fclose(fp, fs, ft); memiliki tiga argumen. Hal ini menyebabkan error kompilasi karena fclose() seharusnya dipanggil secara terpisah untuk setiap file, seperti fclose(fp); fclose(fs); fclose(ft);.



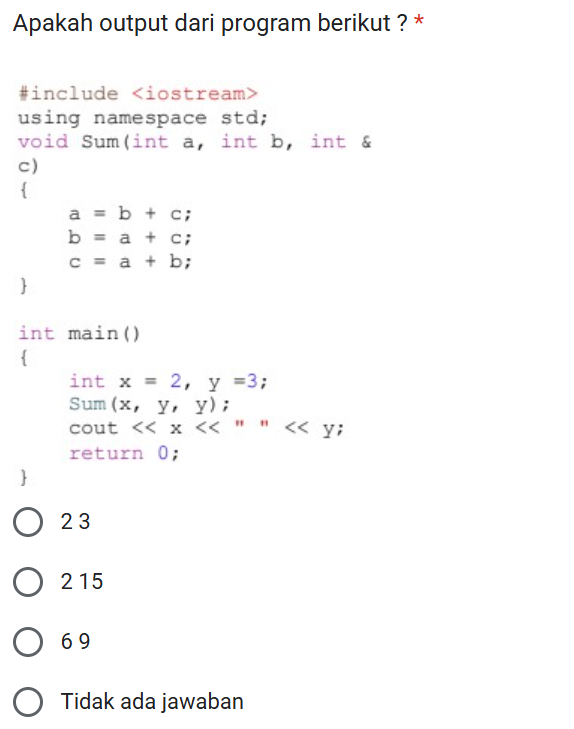
Jawaban yang benar adalah **"Hello Hi"**.

Program memanggil fungsi fun(main), yang menyebabkan eksekusi printf("Hello ");, sehingga mencetak "Hello ". Setelah kembali ke main(), program mengeksekusi printf("Hi\n");, sehingga mencetak "Hi". Jadi, output akhirnya adalah **"Hello Hi"**.



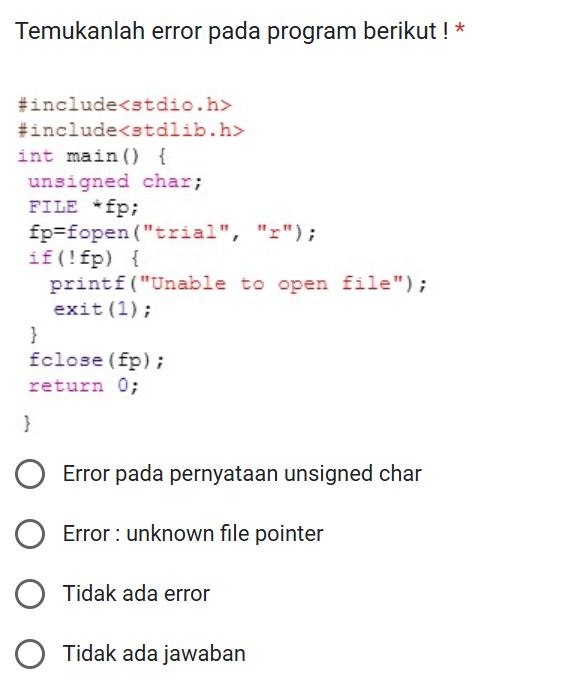
Jawaban yang benar : Output : Unable to open file

Pada path file "c:\tc\file.c", karakter \t di dalam string akan dianggap sebagai escape sequence (tab).



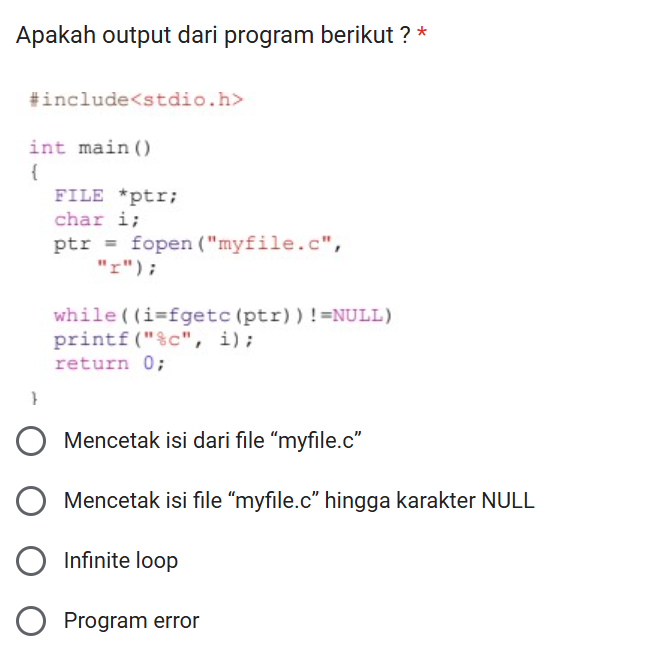
Jawaban yang benar adalah **2 15**.

Penjelasan: Fungsi Sum() menggunakan parameter referensi (int &c), sehingga perubahan pada c akan mempengaruhi variabel asli y di main(). Namun, parameter a dan b hanya dikirimkan sebagai nilai (pass by value), sehingga perubahan pada a dan b di dalam fungsi tidak mempengaruhi variabel x dan y di main(). Setelah eksekusi Sum(x, y, y), nilai y menjadi 15 (karena c = a + b = 2 + 3 = 5, lalu c = a + b = 2 + 5 = 7, dan terakhir c = a + b = 2 + 7 = 15), tetapi x tetap 2. Oleh karena itu, outputnya adalah **"2 15"**.

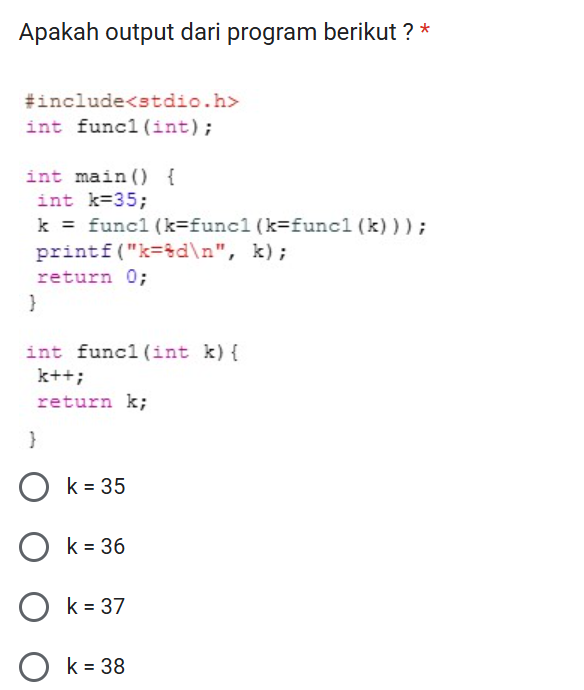


Jawaban yang benar adalah **"Tidak ada error”**

**Penjelasan:**



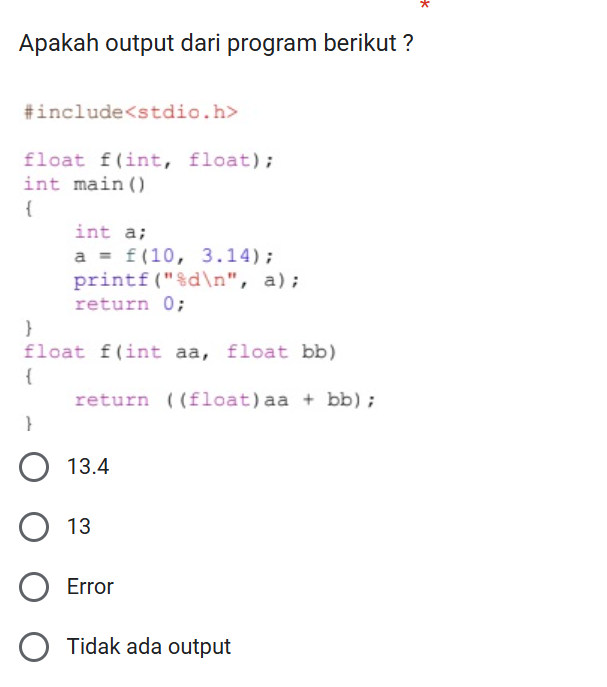
**Jawaban:** Mencetak isi dari file “myfile.c”  
**Penjelasan:**



Jawaban: **k = 38**

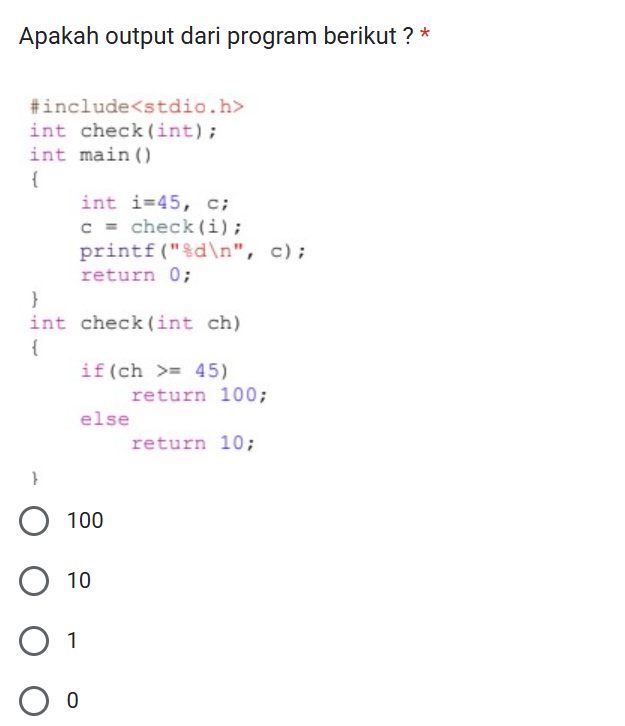
Penjelasan:  
Fungsi func1(int k) meningkatkan nilai k sebanyak **1** setiap kali dipanggil. Dalam ekspresi k = func1(k = func1(k = func1(k))), pemanggilan fungsi berlangsung dari dalam ke luar:

1. func1(35) → 36
2. func1(36) → 37
3. func1(37) → 38  
   Akhirnya, k = 38, sehingga output program adalah **k=38**.



Jawaban: **13**

Penjelasan:  
Fungsi f(int, float) mengembalikan nilai bertipe float, tetapi dalam main(), nilai hasil dari f(10, 3.14) disimpan dalam variabel a yang bertipe int. Saat f(10, 3.14) dieksekusi, hasilnya adalah 10 + 3.14 = 13.14, tetapi karena disimpan dalam a yang bertipe int, angka desimalnya akan **terpotong** (bukan dibulatkan). Oleh karena itu, output yang dicetak adalah **13**.



Jawaban: **100**

Penjelasan:  
Variabel i diinisialisasi dengan nilai 45, lalu dipassing ke fungsi check(int ch). Di dalam fungsi, kondisi if (ch >= 45) akan **benar** karena ch = 45, sehingga fungsi akan mengembalikan nilai 100. Nilai ini kemudian disimpan di c dan dicetak dengan printf("%d\n", c);, sehingga output program adalah **100**.



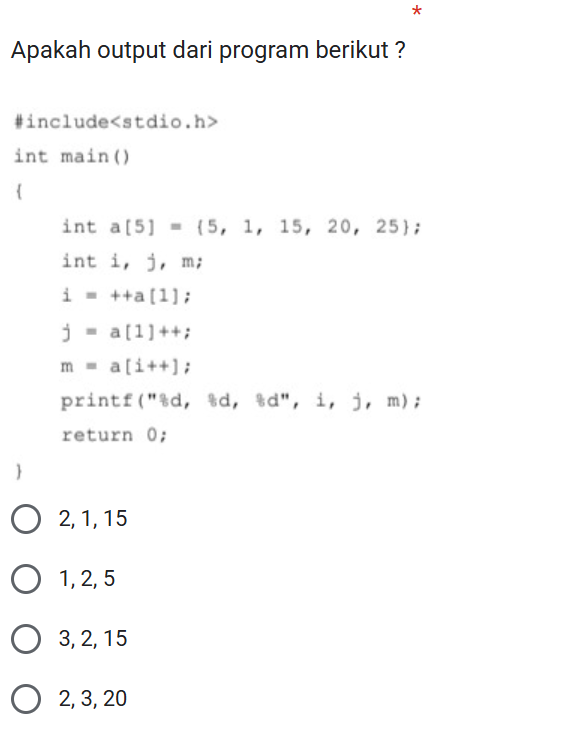
Jawaban: **"Trh"**

Penjelasan:  
Program ini membaca file source.txt yang berisi "To err is human" dan menyalin beberapa karakter ke file target.txt. Dalam loop, setiap karakter dibaca dengan getc(fs), lalu dilakukan fseek(fs, 4L, SEEK\_CUR), yang berarti setelah membaca satu karakter, pointer file akan maju **4 karakter**. Hanya karakter yang dibaca pertama kali dalam setiap iterasi yang akan ditulis ke target.txt.

Mari kita lihat bagaimana karakter dipilih:

1. **Baca 'T'**, lewati **'o', ' ', 'e', 'r'**
2. **Baca 'r'**, lewati **'r', ' ', 'i', 's'**
3. **Baca 'h'**, lewati **'u', 'm', 'a', 'n'**

Jadi, karakter yang masuk ke target.txt adalah "Trh".



Jawaban: **3, 2, 15**

Program ini menggunakan array a[5] = {5, 1, 15, 20, 25}; dan beberapa operasi pada indeksnya. Pertama, i = ++a[1]; meningkatkan a[1] dari 1 menjadi 2 sebelum menyimpannya ke i, sehingga i = 2. Selanjutnya, j = a[1]++; menyimpan nilai awal a[1] (yaitu 2) ke j, lalu meningkatkan a[1] menjadi 3, sehingga j = 2. Terakhir, m = a[i++]; mengambil a[2] (yaitu 15) dan menyimpannya ke m, kemudian meningkatkan i menjadi 3, sehingga m = 15.

Dengan nilai akhir i = 2, j = 3, dan m = 15, maka output yang dicetak adalah **"2, 3, 15"**.