**FINAL PROJECT**

Muhammad Aqil Farrukh

5025221158

**SOAL NOMOR 1**

Berapa banyak bilangan bulat positif berurutan yang dapat Anda temukan, sehingga jumlah digit (dalam representasi desimal) dari masing-masing digit tidak habis dibagi 13? Catatan: Karena 49 adalah bilangan pertama yang jumlah digitnya habis dibagi 13, maka misalnya bilangan bulat dari 1 sampai 48 memenuhi syarat.

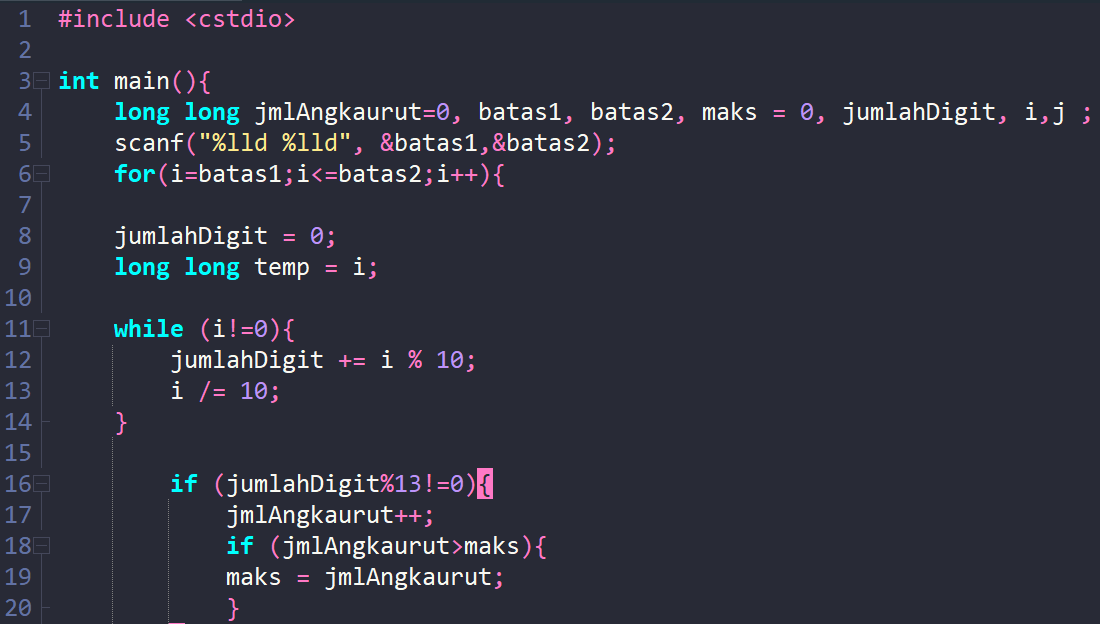
Untuk menyelesaikan permasalahan tersebut yaitu dengan menemukan sebuah keteraturan dan pola terlebih dahulu dari problem ini. Saya telah membuat tabel dari test case awal sampai angka 200-an. Tabel nya adalah sebagai berikut.

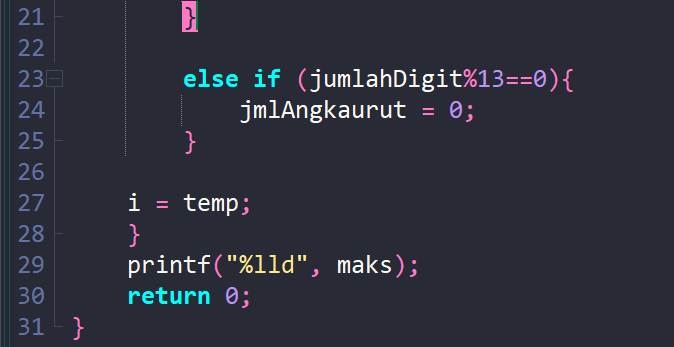


Dari tabel tersebut, saya mendapatkan bahwa nilai maksimal dari jumlah deret berurutan yang tidak bisa dibagi 13 adalah 48. Tetapi setelah saya mencoba submit 48, masih wrong answer. Karena kemungkinan nilainya masih kurang, saya akan mencoba dari 1 sampai batas integer yaitu 2 x 109 .

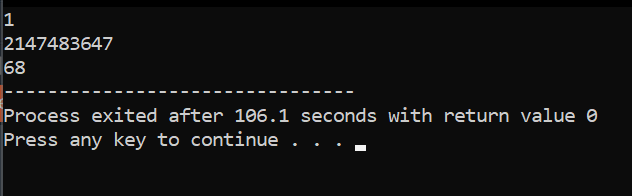
Untuk memudahkan dalam perhitungan, saya telah membuat program untuk menghitung digit terlebih dahulu dari input yang diberikan, kemudian mengecek apakah bisa dibagi dengan 13. Kemudian mengecek nya berturut -turut agar dapat memberikan nilai dari urutan bilangan dengan jumlah yang maksimal dari interval tertentu.

Kodingannya adalah sebagai berikut.





Dengan menggunakan program itu, saya mencoba input dari 1 sampai batas integer yaitu 2,147,483,647 . Hasil yang saya dapatkan adalah sebagai berikut.



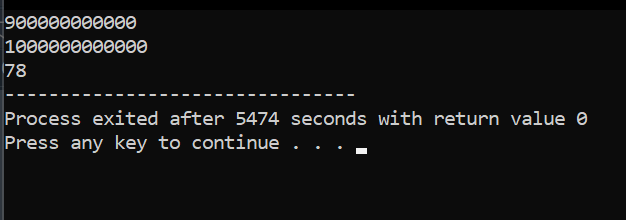
Setelah mendapat hasilnya, ditemukan nilainya adalah 68. Tapi, setelah saya submit ke SPOJ masih menunjukkan wrong answer.

Saya kemudian mencari nilai dari interval yang lebih besar yaitu dari 1 x 10^10 sampai 2x10^10. Dan hasilnya adalah sebagai berikut.

Text

Description automatically generated

Ternyata masih belum berubah, nilai maksnya masih 68. Saya mencoba nilai yang lebih tinggi yaitu 9 x 10^11 sampai dengan 1 x 10^12. Hasilnya adalah sebagai berikut.



Ternyata hasilnya bisa berubah. Nilai maks yang sebelumnya 68 menjadi 78. Setelah saya coba submit, ternyata accepted. Sehingga nilai maks atau banyaknya bilangan bulat positif berurutan dimana tidak bisa dibagi 13, paling banyak adalah 78.

**SOAL NOMOR 2**

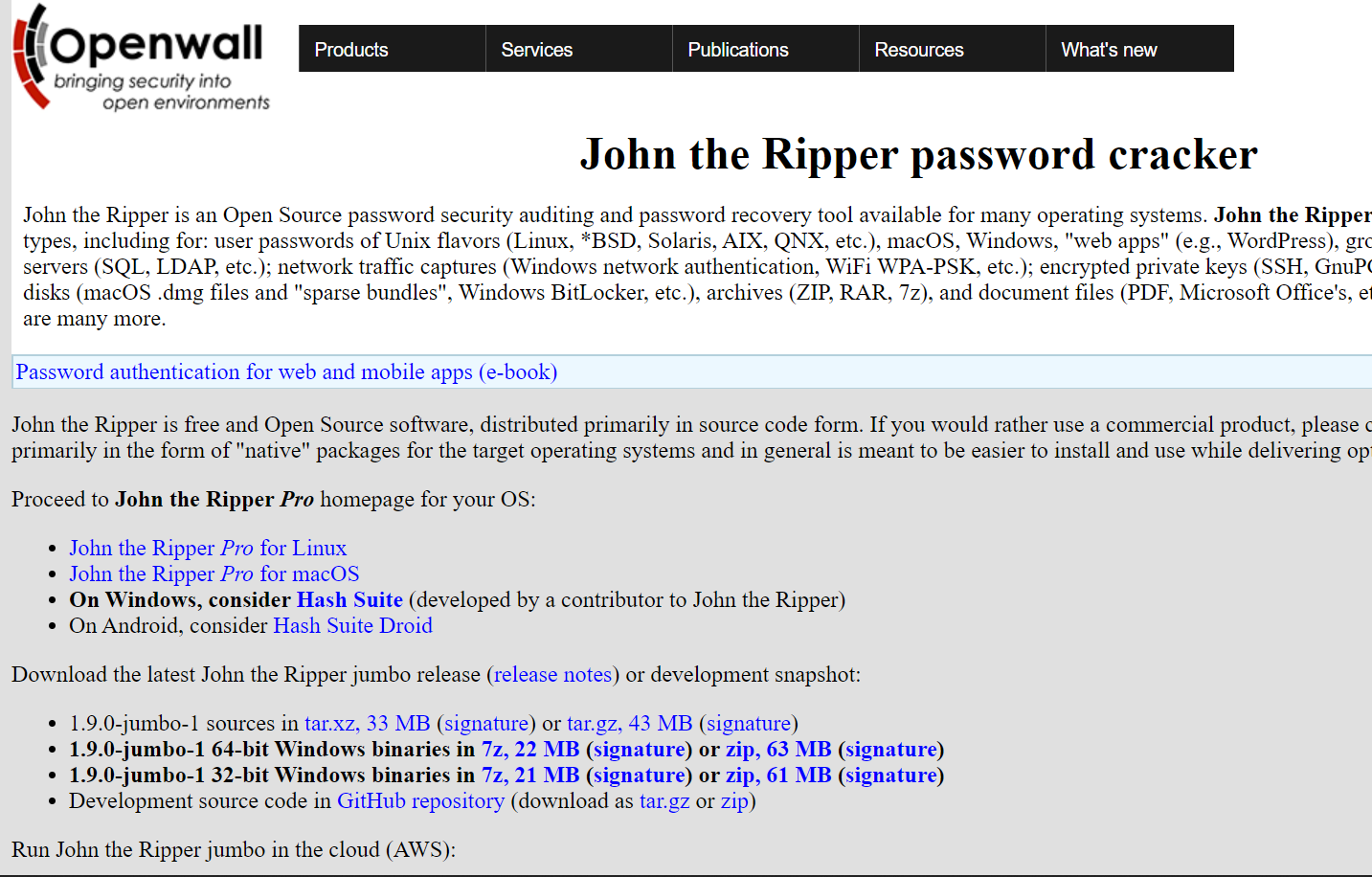


Untuk mengerjakan problem ini, saya awalnya mencari cara di youtube “Cara membuka file zip yang terkunci” Kemudian saya mengetahui bahwa ada tools yang namanya John The Ripper yang dapat digunakan untuk membuka file zip yang terkunci. Sumber video nya adalah sebagai berikut.

<https://www.youtube.com/watch?v=mXWyp9xCaJU&t=680s>

Dari video tersebut saya mencoba mengikuti video dari 10.00 sampai 11.26. Tetapi setelah saya coba, password nya tidak mau muncul. Meskipun begitu, saya mendapatkan file hashya. Cara menggunakan John The Ripper untuk mendapatkan file hashnya adalah sebagai berikut.

1. Download file John The Ripper di websitenya. <https://www.openwall.com/john/> . Kemudian saya pilih yang 1.9.0 Jumbo 64 bit seperti gambar screenshot di bawah.





1. Langkah kedua yaitu meng-ekstrak file download tersebut, dan memindahkan zip yang terkunci ke dalam folder hasil ekstrak yang bertuliskan “run”

Graphical user interface, application

Description automatically generated

1. Setelah memasukkan file answer.zip ke dalam folder run. Buka folder run dan ketik “cmd” seperti gambar berikut.

A screenshot of a computer

Description automatically generated with medium confidence

1. Kemudian akan muncul pop up seperti ini

Text

Description automatically generated

1. Kemudian ketik “ zip2john.exe answer.zip > answer.hash “ . Pengetikan tersebut bertujuan untuk membuka program zip2john yang akan berfungsi untuk membuat hash dari file zip yang akan kita cari passwordnya. Kemudian hash tersebut akan disimpan ke dalam answer.hash. Text

   Description automatically generated
2. Setelah mendapatkan file hash, kemudian hapus bagian yang tidak perlu sehingga hanya hash nya saja yang tersisa.

SEBELUM

Text, application

Description automatically generated

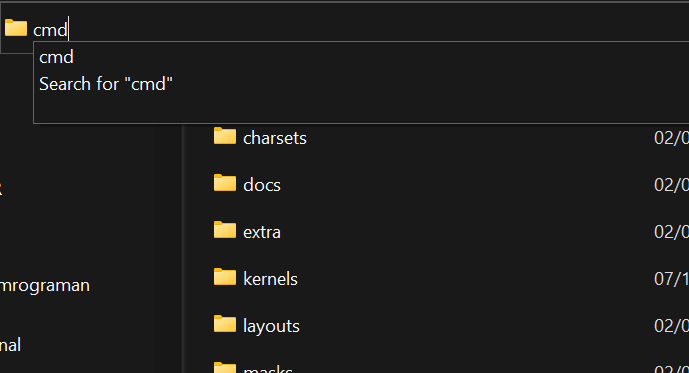
SESUDAH

Text

Description automatically generated

Karena saya tidak dapat menggunakan John The Ripper untuk menganalisis hashnya, saya akan menggunakan hashcat dalam membuka password dari zip tersebut. Langkah – langkah dalam menggunakan hashcat adalah sebagai berikut.

1. Download hashcat terlebih dahulu dari websitenya. <https://hashcat.net/hashcat/> . Pilih yang paling atas yaitu yang binaries.
2. Setelah itu di ekstrak dan file hash yang ada pada folder john dipindahkan ke dalam folder hashcat.
3. Kemudian ketik cmd seperti langkah yang dilakukan sebelumnya pada John The Ripper.



1. Kemudian ketik hashcat –help. Maka akan muncul beberapa daftar yang akan membantu kita dalam kebutuhan untuk mencari password zip.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

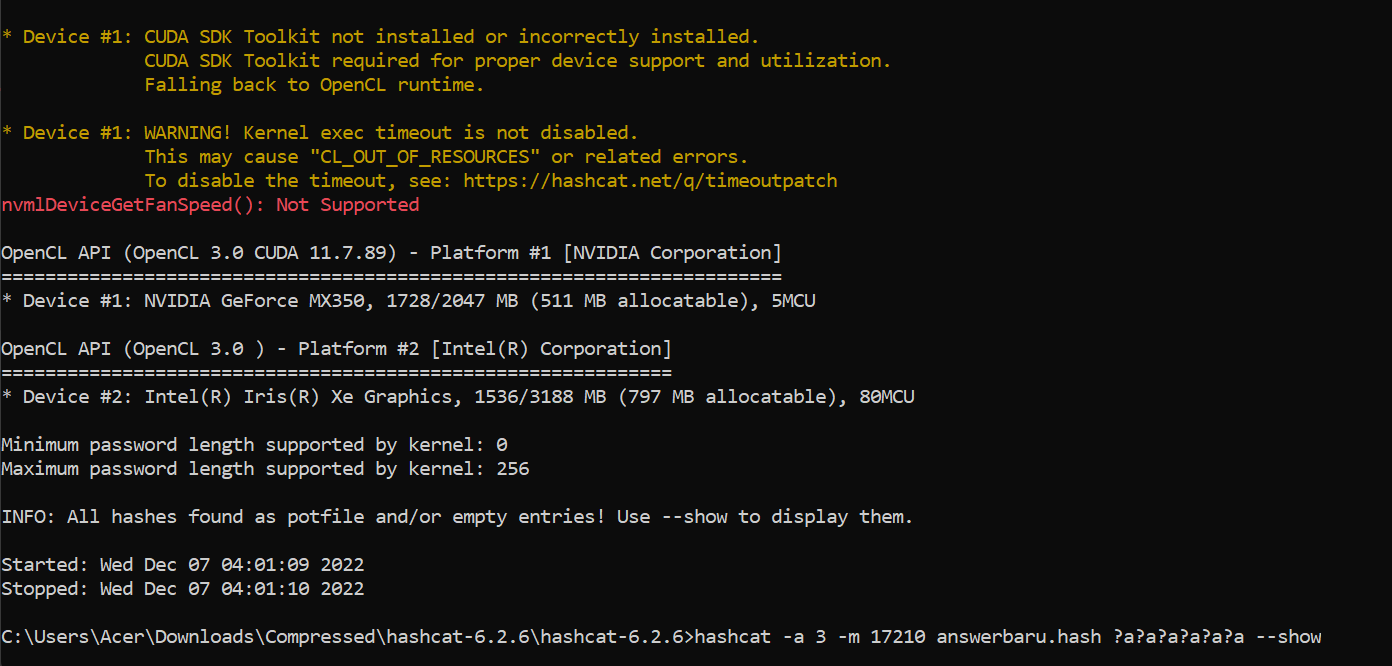
Description automatically generated

1. Dari daftar tersebut , yang akan kita gunakan adalah perintah Attack-Mode brute force. Tapi sebelum itu, kita butuh kode hash yang berkaitan dengan file zip yang akan kita buka. Maka harus di cari kodenya dengan ctrl-F dan ketik zip. Setelah itu akan menemukan kode seperti gambar pada di bawah ini.

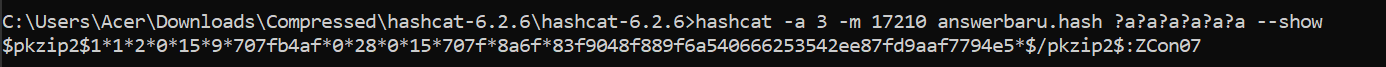
Graphical user interface, text

Description automatically generated

1. Dari pencarian sebelumnya, didapatkan bahwa kode hash untuk PKZIP uncompressed adalah 17210. Kemudian tinggal copy paste command yang ada pada Brute-Force yaitu “ hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a “ dan mengganti angka 0 dengan 17210, dan juga mengganti exmaple0 dengan nama file hash yang sudah disiapkan.



1. Kemudian setelah selesai, ketik kode yang sama tetapi dengan tambahan - - show setelah command yang kita ketik. Seperti pada gambar di atas. Hasilnya adalah sebagai berikut.



Telah didapatkan bahwa password dari Zip tersebut adalah ZCon07. Kemudian file zip diekstrak dengan mengisi password tersebut.

Graphical user interface, text, application

Description automatically generated

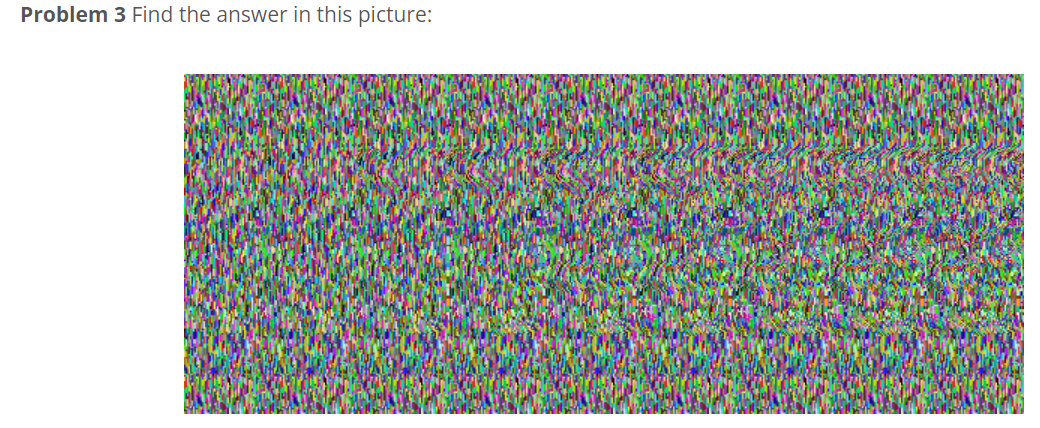
Kemudian isi dari file answer tersebut adalah sebagai berikut.

Graphical user interface, text, application

Description automatically generated

Jadi jawaban untuk nomor 2 adalah 857648936

**SOAL NOMOR 3**



Untuk menyelesaikan project ini, saya memulai dengan melihat gambar terlebih dahulu. Awalnya saya bingung untuk menyelesaikan problem ini, kemudian saya mencoba memandangi gambar ini selama 10 menit. Gambaran awal yang saya lihat adalah gambar kambing. Jadi saya mulai mencoba untuk submit jawaban “Goat” kemudian “Sheep”. Setelah saya mencoba dua jawaban itu, ternyata salah.

Kemudian saya mulai menggunakan google untuk mengetahui jenis gambar tersebut, setelah saya mengetahui bahwa gambar tersebut adalah gambar stereogram. Saya mulai mengikuti arahan dari google untuk mengetahui cara membaca stereogram. Saya mengikuti langkah – langkah yang ada di website <https://www.sepulsa.com/blog/bagaimana-cara-melihat-stereogram> . Tetapi setelah saya coba menyelesaikan dengan itu, saya merasa pusing karena melihat laptop terlalu dekat dalam waktu yang lumayan lama. Jadi saya mulai mencari cara lain yaitu dengan menggunakan website untuk solve stereogram. Website yang muncul di pencarian pertama yaitu <https://piellardj.github.io/stereogram-solver/> . Kemudian langkah saya dalam menyelesaikan problem ini adalah sebagai berikut.

1. Masuk ke dalam website <https://piellardj.github.io/stereogram-solver/> .
2. Kemudian pilih file png stereogram yang ada di soal spoj, dengan cara mendownload nya terlebih dahulu, kemudian di upload ke dalam website tersebut. Cara mengupload nya adalah dengan mengklik tombol “Choose File” seperti gambar berikut.

Text

Description automatically generated

1. Kemudian hasil dari stereogram akan terlihat seperti berikut.



Jadi, jawaban untuk soal ke 3 adalah TARP.

**SOAL NOMOR 4**

Saat melihat bilangan dari kiri ke kanan, jika tidak ada angka yang lebih kecil dari angka di sebelah kirinya, maka bilangan tersebut disebut bertambah atau *“increasing”*; misalnya, 125589.

Demikian pula jika tidak ada angka yang lebih kecil dari angka di sebelah kanannya, angka tersebut disebut menurun atau *decreasing*; misalnya, 995421.

Kami akan memanggil bilangan bulat positif yang tidak menambah atau mengurangi angka "memantul" atau “bouncy” ; misalnya, 64783.

Jelas tidak mungkin ada angka melenting di bawah seratus, tetapi lebih dari setengah angka di bawah seribu (525) melenting. Nyatanya, angka terkecil yang proporsi angka pantulan pertama kali melebihi 50% adalah 538. Angka pantulan menjadi semakin umum dan pada saat kita mencapai 21780 proporsi angka pantulan sama dengan 90%.

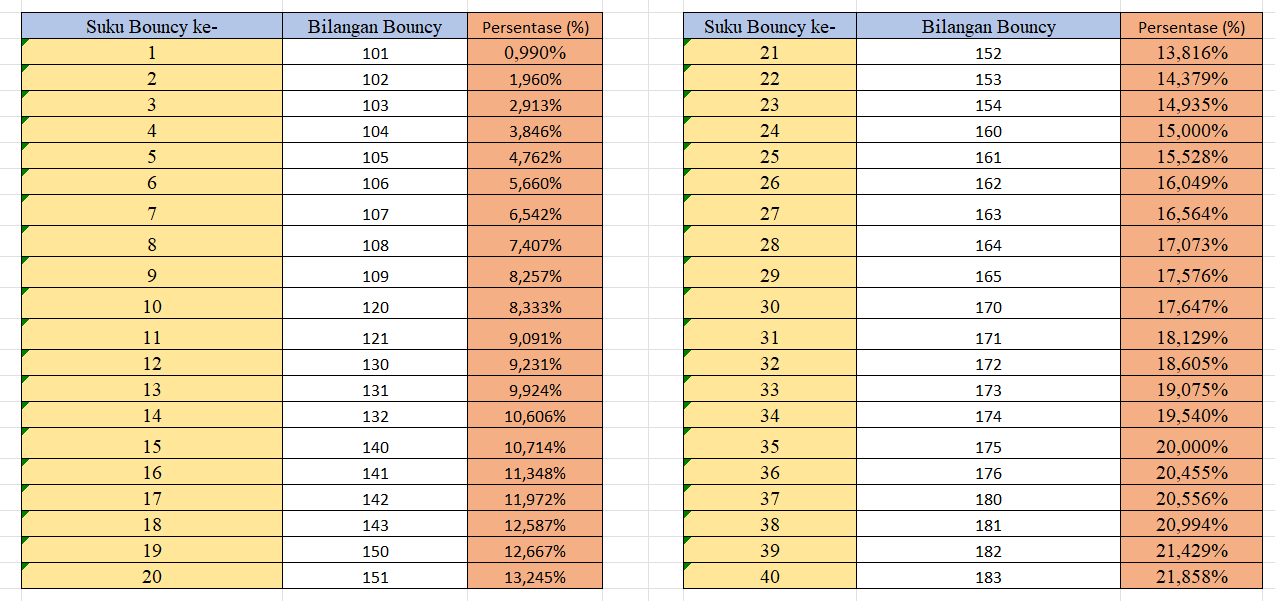
Temukan angka terkecil yang proporsi angka pantulnya persis 99%.

Penjelasan:

Dalam soal ini disebutkan angka-angka tersebut mengalami pemantulan atau bisa disebut mengalami penambahan dan pengurangan dalam setiap digitnya. Karena angka dibawah 100 tidak ada yang memantul. Maka mulai memantul dari angka 101.

Karna mulai dari angka 101, jadi dapat disimpulkan bahwa 101 memiliki persentase memantul 0%. Lalu alasan mengapa 101 adalah 0% karena 101 adalah suku pertama dari bounce atau angka memantul pertama. Apabila 1 dibagi dengan 101 akan menghasilkan 0,0099. Apabila hasil tersebut di kali kan dengan 100% akan menghasilkan 0,99%.

Dari analisis tersebut, akan dibuat beberapa testcase dengan tabel yang berisi dari suku pertama sampai suku tertentu. Tabel nya adalah sebagai berikut.



Dari tabel tersebut dapat diketahui bahwa ada nilai persen yang hampir sama untuk setiap suku. Contohnya adalah suku ke-9 dan suku ke-10. Apabila akan dicari persentase 8%, maka kemungkinan besar yang akan muncul adalah angka 109. Karena nilai persentase nya yang paling mendekati 8%. Apabila ingin mencari nilai dari 120, maka jika dibuat program, harus menginput 83/1000 atau 8,3/100, dengan begitu nilai yang akan muncul adalah 120 bukan 109.

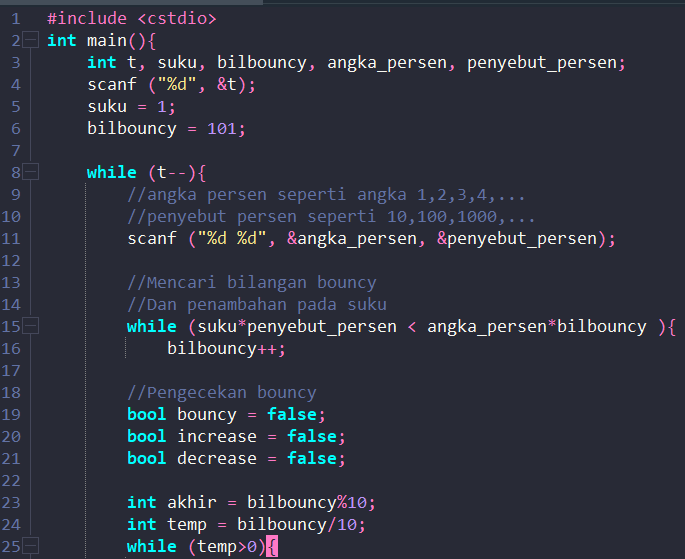
Dari tabel tersebut, sudah didapatkan logika untuk membuat program yang akan mengecek bilangan bouncy. Dimana program ini akan dimulai dari angka 100 ke atas, karena dibawah 100 tidak ada angka yang bouncy. Program yang akan dibuat, akan mengecek nilai bouncy sampai pembagian antara suku dan bilangan bouncy mendekati persentase yang kita inginkan. Tetapi karena ada masalah dalam pendekatan nilai desimal, seperti suku ke-9 dan 10 sebelumnya, operasi pembagian yang ada di program akan dibuat menjadi perkalian saja yang paling mendekati. Rumus awal nya adalah sebagai berikut.

Lalu diuraikan menjadi,

Apabila diubah menjadi perkalian maka,

Jadi nilai dari suku/bilangan harus lebih dari nilai dari persentase yang diinginkan. Tetapi apabila rumus tersebut dimasukkan program, dan dijadikan perulangan, maka tanda harus dibalik menjadi agar nilai yang diinginkan dicari terus sampai melebihi ruas kanan.

Program untuk mengecek bouncy dan persentase adalah sebagai berikut.



Text

Description automatically generated

Kemudian dengan program tersebut, diinputkan nilai 99% dan menghasilkan nilai sebagai berikut.

Graphical user interface, text

Description automatically generated

Jadi angka bouncy yang proporsinya tepat 99% adalah 1587000

**SOAL NOMOR 5**

Radikal dari n, rad(n) - adalah hasil kali faktor prima berbeda dari n. Misalnya, 1008 = 2^4\*3^2\*7 , jadi rad(1008) = 2\*3\*7 = 42.

Jika kita menghitung rad(n) untuk 1 <= n <= 10, lalu mengurutkannya berdasarkan rad(n), memutuskan hubungan dengan menyortir berdasarkan nilai n, kita mendapatkan:

Tidak disortir:

n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

rad(n) = 1, 2, 3, 2, 5, 6, 7, 2, 3, 10

Diurutkan:

n = 1, 2, 4, 8, 3, 9, 5, 6, 7, 10

rad(n) = 1, 2, 2, 2, 3, 3, 5, 6, 7, 10

k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Misalkan E(k) adalah elemen ke-k dalam kolom n yang diurutkan; misalnya, E(4) = 8 dan E(6) = 9.

Jika rad(n) diurutkan untuk 1 <= n <= 100000, cari E(10000).

Penjelasan :

Untuk menyelesaikan problem ini, tinggal urutkan hal-hal yang harus dilakukan terlebih dahulu untuk menyelesaikan problem ini. Saya berencana untuk menyelesaikan problem ini langsung menggunakan program. Karena logika dari soal ini terhitung mudah untuk dicerna menurut saya, mungkin yang akan sedikit sulit adalah kodingannya. Langkah-langkah tersebut adalah sebagai berikut.

1. Saya akan mencari angka prima atau generate angka prima dari 1 sampai 100.000. Saya menggunakan angka 100.000 karena yang ditanyakan pada soal ini adalah 10.000. Seharusnya 100.000 sudah memadai untuk digunakan sebagai batas dalam program yang akan saya gunakan. Karena saya sudah pernah berlatih di spoj untuk menyelesaikan sebuah problem yang berjudul “Prime Generator” dengan kode “PRIME1”. Maka dapat digunakan codingannya dengan sedikit modifikasi. Berikut adalah bukti acc Prime1 dan codingan yang sebelum dimodif dan setelah di modif



SEBELUM MODIF

Text

Description automatically generated

SETELAH DIMODIF

Text

Description automatically generated

1. Kemudian bilangan prima yang telah saya generate akan saya simpan dalam sebuah array seperti gambar di atas. Dan akan saya sort menggunakan bubble sort.

Text

Description automatically generated

1. Kemudian dengan menggunakan perulangan, saya akan generate angka biasa dari 1-100.000 yang akan di simpan di array. Di dalam perulangan i tersebut, saya akan membuat perulangan yang kedua untuk mengecek apakah bilangan i tersebut bisa dibagi dengan bilangan prima yang telah disortir dengan menggunakan array juga. Apabila angka i tersebut bisa dibagi dengan prima, maka faktor prima nya akan dikalikan dan disimpan di array. Kodingan untuk langkah ke-3 ini adalah sebagai berikut.

Text

Description automatically generated

1. Setelah perkalian faktor tersebut saya kalikan dan simpan di array. Saya akan sortir itu dengan bubble sort. Yang akan tersortir bukan hanya perkalian faktor saja, melainkan angka [ ] yang sebelumnya digunakan untuk menyimpan angka dari 1-100.000 pun akan di sortir mengikuti perkalian angka sebelumnya. Apabila posisi perkalian angka berubah, maka angka[j] juga akan berubah posisi dan urutannya. Bukti kodingan untuk langkah ini adalah sebagai berikut.

Text

Description automatically generated

1. Setelah semua codingan tersebut disusun, selanjutnya adalah tinggal mencari nilai dari E(10.000) . Bukti keseluruhan codingan untuk problem ini beserta hasilnya adalah seperti berikut.

Text

Description automatically generated

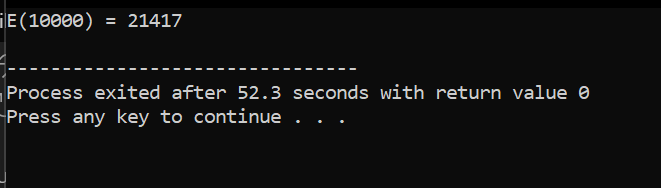
Text

Description automatically generated

Text

Description automatically generated

Codingan tersebut saya buat sendiri, meski kodingan tersebut terdiri sampai 70 lines. Tetapi kodingan tersebut disusun secara terstruktur berdasarkan langkah-langkah yang saya gunakan dalam mengerjakan problem ini. Hasilnya adalah sebagai berikut.



Jadi jawaban untuk nomor 5 adalah 21417.

**SOAL NOMOR 8**

Berapa banyak cara dapat 50! dinyatakan sebagai jumlah dari dua atau lebih bilangan bulat positif berurutan?

Penyelesaian :

Problem tersebut memiliki soal dengan angka faktorial yang sangat besar. Sehingga saya akan mencoba beberapa test case dengan angka faktorial yang kecil dan menganalisa pola keteraturannya terlebih dahulu. Dalam percobaan ini, saya mencoba mengetik sendiri di excel dan menghitung sendiri. Tabel hasil perhitungan yang saya dapatkan adalah sebagai berikut.

Table

Description automatically generated

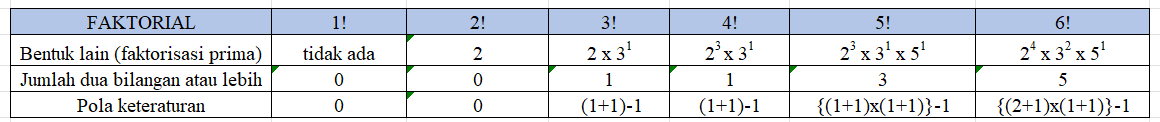
Dari perhitungan yang telah saya lakukan, saya dapat membuktikan jumlah tersebut dengan cara membagi hasil faktorial dengan faktor prima ganjil.

Contoh nya saya membagi 3! dengan faktor prima nya yaitu 3. Maka akan didapatkan hasil 2. Maka dari itu penjumlahannya adalah 1+2+3 dengan posisi 2 ada di tengah. Hal ini berlaku untuk pembagian faktorial setelahnya.

Dari perhitungan yang telah saya lakukan juga, saya mendapatkan bahwa nilai hasil faktorial tidak dapat di jadikan jumlah dua bilangan atau lebih karena akan menghasilkan nilai yang genap dimana tidak ada nilai tengah yang berbentuk bilangan bulat.

Contoh nya 3! Apabila saya bagi dengan 2 akan menghasilkan nilai 3. Angka 2 tidak memiliki nilai tengah yang berupa bilangan bulat, melainkan angka koma. Apabila jumlah suku sebuah deret adalah 2, maka nilai tengah dari deret itu adalah (suku ke-1 + suku ke-2) dibagi dengan 2. Yang mana akan menghasilkan nilai koma apabila bilangan dalam suku tersebut berurutan. Maka agar dapat mendapatkan nilai dari jumlah dua bilangan, nilai hasil faktorial harus dibagi dengan faktor prima ganjil nya atau perkalian antar faktor prima ganjilnya.

Dari tabel tersebut, ditemukan sebuah keteraturan pola. Dimana pola tersebut adalah hubungan antara faktor prima dan jumlah dua bilangan atau lebih. Karena angka dua tidak akan menghasilkan nilai bukti jumlah, dianggap angka dua dan perpangkatannya diabaikan karena tidak mempengaruhi hasil. Pola keteraturan dari pola tersebut akan diungkap di tabel berikut.



Dari pola keteraturan tersebut didapatkan bahwa setiap nilai pangkat yang ada pada faktor prima selain angka 2 akan ditambah 1 dan dikalikan dengan setiap pangkat yang lain. Kemudian akan dikurangi dengan 1. Hasil pengurangan tersebut menunjukkan jumlah dua bilangan atau lebih yang dapat menghasilkan nilai faktorial.

Karena perintah soal adalah mencari nilai dari penjumlahan dua buah bilangan atau lebih dari 50!. Langkah pertama yaitu adalah mencari nilai faktor prima dari 50!. Karena itu adalah bilangan yang sangat besar dan akan sangat lama dalam menghitungnya, saya mengambil cara yang ada pada website <https://janmr.com/blog/2010/10/prime-factors-of-factorial-numbers/> .

Di website tersebut diajarkan untuk mencari faktor prima dari sebuah nilai faktorial. Langkah langkah nya adalah sebagai berikut.

* Mencari 2 pangkat n.
* Mencari 3 pangkat n
* Mencari 5 pangkat n.
* Mencari 7 pangkat n
* Mencari 11 pangkat n.
* Dan prima seterusnya sampai 47 pangkat n.

Alasan kenapa hanya sampai 47 pangkat n karena 50 masih bisa dibagi 47. Sedangkan prima setelah 47 yaitu 53, apabila 50 dibagi 53 tidak akan ada yang bisa dibagi. Untuk memudahkan perhitungan, saya akan akan menggunakan program saja dalam pembagiannya. Kodingannya adalah sebagai berikut.

Text

Description automatically generated

Text

Description automatically generated

Karena yang ditanyakan di problem ini adalah ada berapa cara untuk membuat 50! Menjadi penjumlahan dua bilangan atau lebih. Maka kodingan untuk mencari pangkat prima sebelumnya, akan saya modif lagi menjadi kodingan yang langsung menghitung ke hasil akhirnya. Kodingannya adalah sebagai berikut.

Text

Description automatically generated

Text

Description automatically generated

Jadi banyak cara agar 50! dinyatakan sebagai jumlah dari dua atau lebih bilangan bulat positif berurutan adalah 93000959.

**SOAL NOMOR 9**

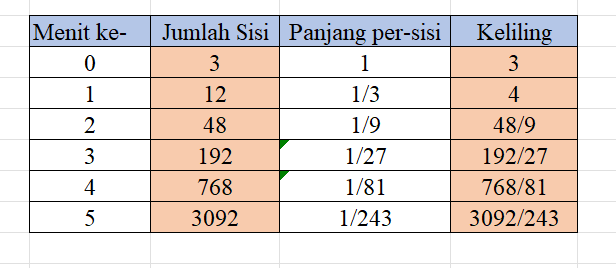
Bayangkan Anda memiliki kristal berbentuk segitiga sama sisi, dengan panjang satu unit di setiap sisinya. Dalam kondisi yang tepat, kristal mulai tumbuh. Setelah satu menit, dua sisi tumbuh dari setiap sisi segitiga yang simetris sempurna. Hasilnya adalah bintang berujung enam, yang memiliki sisi yang persis 1/3 dari panjang sisi asalnya. Setelah satu menit, setiap sisi menumbuhkan dua sisi lagi yang persis 1/3 dari panjang sisi asalnya.

A picture containing shape

Description automatically generated

Tantangan Anda adalah menemukan keliling (dibulatkan ke bilangan bulat terdekat) setelah satu jam tiga puluh tiga menit.

Untuk menyelesaikan problem ini, dapat dianalisis terlebih dahulu pola keteraturan dari setiap penambahan sisi, menit, dan rasio panjang setiap sisi. Pola yang ada pada gambar tersebut dapat dinyatakan seperti pada tabel berikut.



Tabel tersebut saya buat di excel dan saya screenshot. Pola itu dapat di analisis sebagai berikut.

Table

Description automatically generated

Sehingga rumus persamaan untuk mencari keliling dari problem diatas adalah

Keliling = pola jumlah sisi x pola panjang

Keliling =

Keliling = (n adalah menit ke-n)

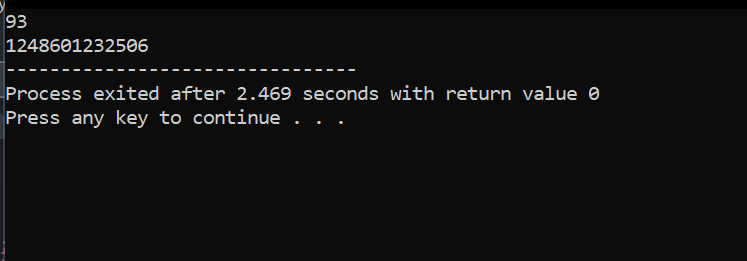
Pertanyaan yang ada di problem tersebut adalah jika waktu yang telah ditempuh adalah 1 jam 33 menit atau 93 menit, berapa kelilingnya?

Maka jawabannya adalah = = 1248601232506

Kodingan untuk menyelesaikan problem ini adalah sebagai berikut.

Text

Description automatically generated



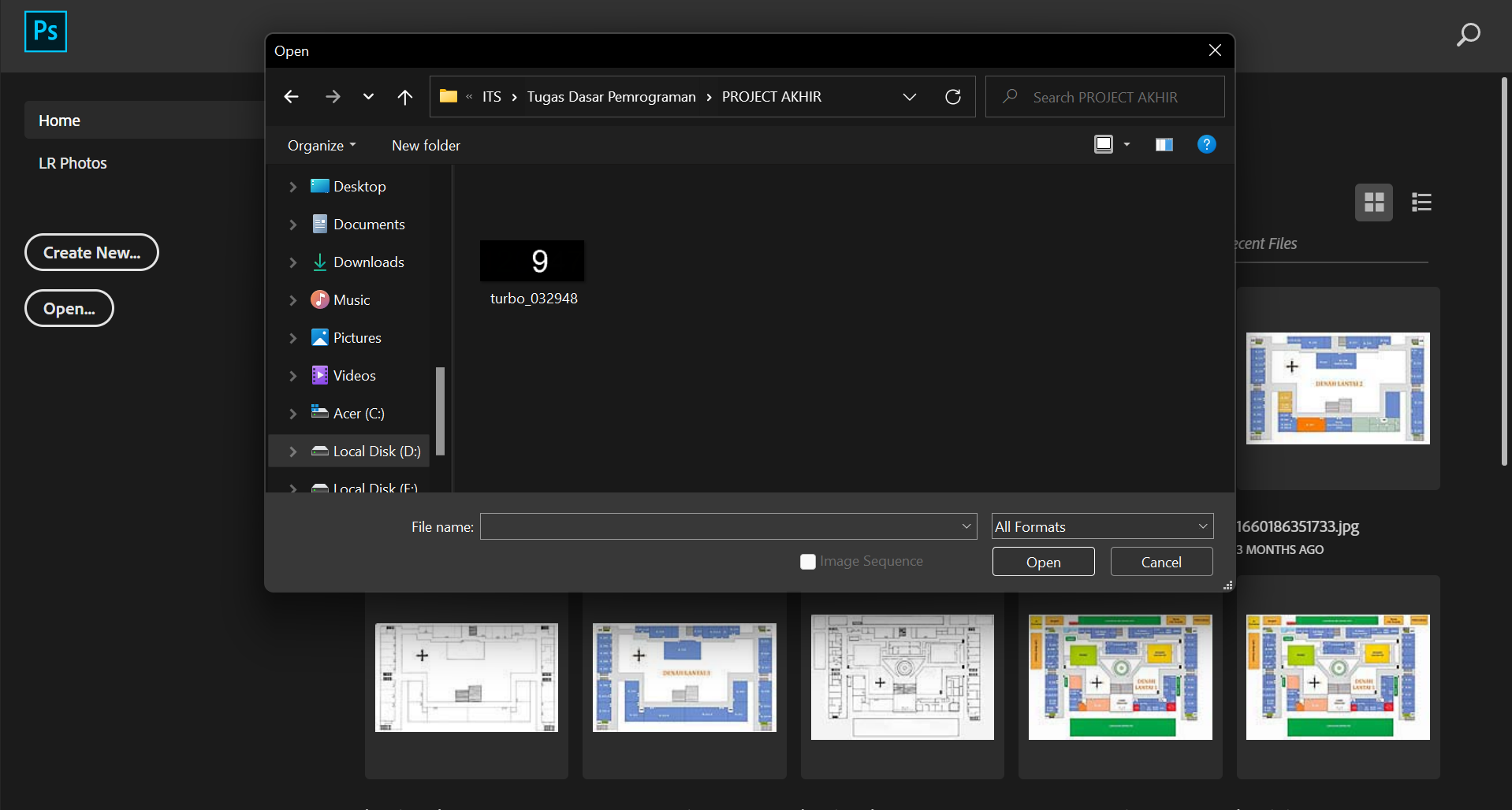
**SOAL NOMOR 10**

Icon

Description automatically generated

Untuk menyelesaikan problem nomor 10 ini, saya mengguanakan tool editing foto yaitu ADOBE PHOTOSHOP CC 2019. Langkah – langkah dalam menyelesaikan problem ini adalah sebagai berikut.

1. Download file foto dari spoj terlebih dahulu, file itu tipe nya adalah gif.
2. Kemudian buka photoshop dan tampilannya sebagai berikut. Kemudian klik open dan pilih file yang akan di upload.



1. Kemudian, karena file nya berbentuk gif, akan diubah dulu ke dalam bentuk png dengan cara di eksport terlebih dahulu, dengan memilih eksport di bagian “file” dan memilih eksport as png.

Graphical user interface, text, application

Description automatically generated

1. Kemudian langkah 1 diulangi , dan memilih file png yang baru saja telah di eksport agar bisa di edit.
2. Lalu ditambah efek saturasi dan brightness di bagian pojok kiri bawah. Seperti gambar berikut



1. Kemudian setelah menambahkan efek saturation dan brighness contrast. Disesuaikan efek saturasinya , brightness dan contrast. Seperti gambar berikut.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Icon

Description automatically generated with medium confidence

Jadi ditemukanlah jawaban untuk problem nomor 10 adalah 120947.

Dari semua penjelasan di atas, saya telah AC sebanyak 8 nomor. Bukti verdict benar 8 adalah sebagai berikut.

