LAPORAN PEKERJAAN RUMAH 01
PERANCANGAN DAN ANALISIS ALGORITMA (B)

Disusun Oleh :

05111940000083 Fajar Satria

Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya
2021

PERMASALAHAN 1 – DIVISIBLE BY 8

Given big integer **n**. Can you rearrange its digits so that the resulting number is divisible by **8**? If there are multiple answers, print the smallest. Do not print the leading '**0**'.

SOLUSI

Dengan bekal membaca Divisibility Ruler, didapatkan bahwa suatu bilangan habis dibagi 8 jika hanya jika 3 digit terakhir dari bilangan tersebut habis dibagi 8. Pendekatan yang saya ambil adalah dengan menyimpan tiap angka dari input ke dalam array of integer, lalu dilanjutkan dengan pengecekan tiap digit yang dibutuhkan oleh bilangan bilangan yang habis dibagi 8. Lalu menyimpan tiap tiap array of integer kedalam array of string secara ascending order dengan aturan bilangan 0 tidak boleh ada didepan. Setelah selesai mengiterasi dan menghasilkan beberapa kemungkinan yang sudah disimpan di array of string, yang harus dilakukan selanjutnya adalah membandingkan tiap tiap string dan mencari string yang berisi angka yang paling minimum.

Input	Array Of Integer	3 digit yang memungkinkan	Bentuk akhir	Output
4241000	array[10] = {3,1,1,0,2,0,0,0,0,0}	000	1244000	1000424
		024	1004024	
		040	1024040	
		104	2004104	
		200	1044200	
		424	1000424	
		400	1024400	
		dst...	dst..	

Tabel 1.1 : Contoh penyelesaian dengan input “4241000”

PSEUDOCODE

```
SOLVE(A)
1.  len = A.length
2.  if len < 3
3.          if (A % 8 == 0)
4.                  return A
5.          else    reverse A
6.                  if (A % 8 == 0)
7.                          return A
8.                  else    return -1
9.  else
10.         let B[0 .. 10] be a new array
11.         //save count every digit into array
12.         for i=0 to len-1
13.                 B[A[i]]++
14.
15.         first_digit = 0
16.         //first digit rule, no leading zero
17.         if len > 3
18.                 for i=1 to 9
19.                         if B[i] > 0
20.                                 first_digit = i
21.                                 B[i]—
22.                                 Break
23.
```

```
//forms the last 3 digits which are divisible by 8
let C[0 .. 1000] be a new array
for i=0 to 992 ,  i=i+8

    //save count every digit into array
    let D[0 .. 1000] be a new array
    dup = i
    D[dup % 10]++
    dup = dup / 10
    D[dup % 10]++
    dup = dup / 10
    D[dup % 10]++

    //add temp array
    let E[0 .. len] be a new array
    E = B

    //save last 3 digit divisible by 8
    last3 = 0
    last2 = 0
    last1 = 0;

    dup = i
    last3 = dup%10
    if ( D[last3] > E[last3])
        continue
    else    E[last3]--
        D[last3]--
    dup = dup / 10;

    last2 = dup%10
    if ( D[last2] > E[last2])
        continue
    else    E[last2]--
        D[last2]--
    dup = dup / 10;

    last1 = dup%10
    if ( D[last1] > E[last1])
        continue
    else    E[last1]--
        D[last1]--

    //Sort ascending order array of integer then save to array of string
    if first_digit == 0 and last1 == 0
        continue
    if first_digit > 0)
        C[m][0] = first_digit
    for j=0 to 9
        for k=0 to E[j] - 1
            C[m][k] = j
    C[m][++k] = last1
    C[m][++k] = last2
    C[m][++k] = last3
    m++;
```

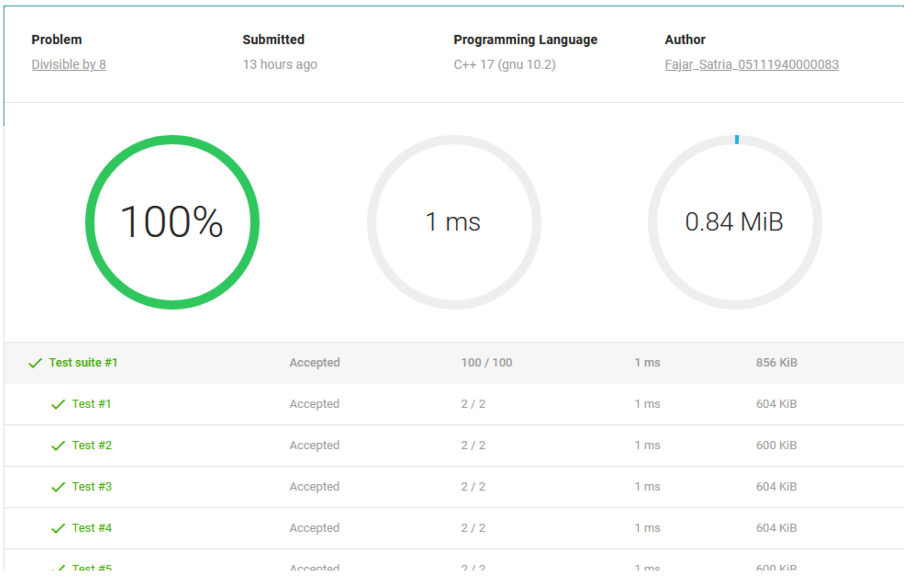
```
        m++;

    if m == 0
        return -1
    else
        //Find minimum number from array of string
        minimum = C[0];
        for i=1 to m-1
            if C[i] < minimum
                minimum = C[i];
        return minimum
```

ACCEPTED SOURCECODE

```
#include <bits/stdc++.h>
using namespace std;
int main(){
	ios_base::sync_with_stdio(false);cin.tie(NULL);
	string number;
	cin >> number;
	int len = number.length();

	if(len<3){
		if(stoi(number) % 8 == 0){
			cout << number << "\n";
			return 0;
		}
		reverse(number.begin(), number.end());
		if (stoi(number) % 8 == 0){
			cout << number << "\n";
			return 0;
		}
		cout << "-1\n";
		return 0;
	}
	int simpanan[10] = {0};
	for(int i=0;i<len;i++)
		simpanan[number[i]-'0']++;
	int awal=0;
	if(len>3){
		for(int i=1;i<=9;i++){
			if(simpanan[i]>0){
				awal=i;
				simpanan[i]--;
				break;
			}
		}
	}
	string kemungkinan[1000];
	int l=0;
	for(int i=0;i<1000;i+=8){
		int dup = i+10000;
		int bagi8[10]={0};
		bagi8[dup % 10]++;
		dup = dup / 10;
		bagi8[dup % 10]++;
		dup = dup / 10;
		bagi8[dup % 10]++;

		int temp_simpan[10]={0};
		copy(begin(simpanan),end(simpanan),begin(temp_simpan));

		int last3=0,last2=0,last1=0;

		dup = i+10000;
		if(bagi8[dup % 10] > temp_simpan[dup % 10]) continue;
```

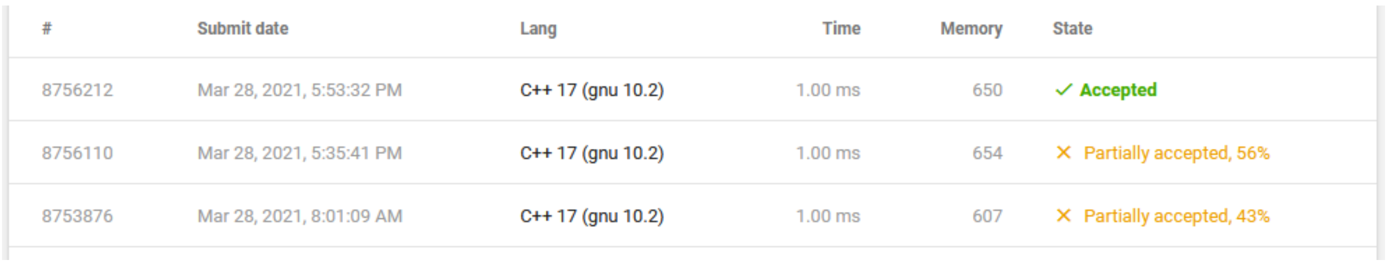
```
            temp_simpan[dup%10]--;bagi8[dup%10]--;
            last3=dup%10;
            dup = dup / 10;
            if(bagi8[dup % 10] > temp_simpan[dup % 10]) continue;
            temp_simpan[dup%10]--;bagi8[dup%10]--;
            last2=dup%10;
            dup = dup / 10;
            if(bagi8[dup % 10] > temp_simpan[dup % 10]) continue;
            temp_simpan[dup%10]--;bagi8[dup%10]--;
            last1=dup%10;

            if(awal==0 && last1==0) continue;
            if(awal>0)
                  kemungkinan[l] += to_string(awal);
            for(int j=0;j<=9;j++){
                  for(int k=0;k<temp_simpan[j];k++){
                        kemungkinan[l] += to_string(j);
                  }
            }
            kemungkinan[l] += to_string(last1);
            kemungkinan[l] += to_string(last2);
            kemungkinan[l] += to_string(last3);
            l++;
      }
      if(l==0){
            cout << "-1\n";
      }else{
            string minimum=kemungkinan[0];
            for(int i=1;i<l;i++)
                  if(kemungkinan[i]<minimum)
                        minimum=kemungkinan[i];
            cout << minimum << "\n";
      }
    return 0;
}
```

SCREENSHOT SUBMISSION



Problem	Submitted	Programming Language	Author
Divisible by 8	13 hours ago	C++ 17 (gnu 10.2)	Fajar_Satria_05111940000083

✓ Test suite #1	Accepted	100 / 100	1 ms	856 KiB
✓ Test #1	Accepted	2 / 2	1 ms	604 KiB
✓ Test #2	Accepted	2 / 2	1 ms	600 KiB
✓ Test #3	Accepted	2 / 2	1 ms	604 KiB
✓ Test #4	Accepted	2 / 2	1 ms	604 KiB
✓ Test #5	Accepted	2 / 2	1 ms	600 KiB



#	Submit date	Lang	Time	Memory	State
8756212	Mar 28, 2021, 5:53:32 PM	C++ 17 (gnu 10.2)	1.00 ms	650	✓ Accepted
8756110	Mar 28, 2021, 5:35:41 PM	C++ 17 (gnu 10.2)	1.00 ms	654	✕ Partially accepted, 56%
8753876	Mar 28, 2021, 8:01:09 AM	C++ 17 (gnu 10.2)	1.00 ms	607	✕ Partially accepted, 43%

Gambar 1.1: *Screenshot submissions*